

Approximation Schemes for Binary Quadratic Programming Problems with Low cp-Rank Decompositions

Khaled Elbassioni* Trung Thanh Nguyen†

November 20, 2014

Abstract

Binary quadratic programming problems have attracted much attention in the last few decades due to their potential applications. This type of problems are NP-hard in general, and still considered a challenge in the design of efficient approximation algorithms for their solutions. The purpose of this paper is to investigate the approximability for a class of such problems where the constraint matrices are *completely positive* and have low *cp-rank*. In the first part of the paper, we show that a completely positive rational factorization of such matrices can be computed in polynomial time, within any desired accuracy. We next consider binary quadratic programming problems of the following form: Given matrices $Q_1, \dots, Q_n \in \mathbb{R}_+^{n \times n}$, and a system of m constraints $x^T Q_i x \leq C_i^2$ ($x^T Q_i x \geq C_i^2$), $i = 1, \dots, m$, we seek to find a vector $x^* \in \{0, 1\}^n$ that maximizes (minimizes) a given function f . This class of problems generalizes many fundamental problems in discrete optimization such as packing and covering integer programs/knapsack problems, quadratic knapsack problems, submodular maximization, etc. We consider the case when m and the cp-ranks of the matrices Q_i are bounded by a constant.

Our approximation results for the maximization problem are as follows. For the case when the objective function is nonnegative submodular, we give an $(1/4 - \epsilon)$ -approximation algorithm, for any $\epsilon > 0$; when the function f is linear, we present a PTAS. We next extend our PTAS result to a wider class of non-linear objective functions including quadratic functions, multiplicative functions, and sum-of-ratio functions. The minimization problem seems to be much harder due to the fact that the relaxation is *not* convex. For this case, we give a QPTAS for $m = 1$.

*Department of Electrical Engineering and Computer Science, Masdar Institute of Science and Technology, Abu Dhabi, UAE

†Department of Electrical Engineering and Computer Science, Masdar Institute of Science and Technology, Abu Dhabi, UAE

1 Introduction

Binary quadratic programming is a classical class of combinatorial optimization problems containing a quadratic function either in its objective or in its constraints. The study of this problem has been of great theoretical interest in the last few decades due to its wide range of applications, for example, capital budgeting and financial analysis [Lau70, MY80], machine scheduling [AKA94], and traffic message management problems [GHS80]. One can mainly distinguish two types of special classes of binary quadratic programming that have been studied extensively in the literature. The first class involves maximizing a quadratic function without any structural constraints (a.k.a *unconstrained 0-1 quadratic problem*) which is known to be APX-hard since it generalizes, for example, the maximum cut problem. The second class deals with the *quadratic knapsack problem* (QKP) arising from the classical knapsack problem by maximizing a quadratic objective function subject to (linear non-negative) knapsack constraint(s) (see. e.g., [Pis07]). Very recently, Yu and Chau [YC13] introduced a new variant of the knapsack problem, called *the complex-demand knapsack problem* (CKP), that arose from the allocation of power in AC (alternating current) electrical systems. In this setting, the constraint is exactly a sum of two squares of linear non-negative functions, and thus is a generalization of the classical knapsack constraint.

Binary programming with quadratic objective function and/or quadratic constraints is hard to approximate in general. For example, quadratic knapsack problem is NP-hard to approximate to within any finite worst case factor [JW02]. Most algorithms for this kind of problems focus on handling various special cases of the general problem [JW02, PS13, KS10, KS12, Xu12]. Those papers investigate the existence of approximation schemes by exploiting the graph-theoretic structure of QKPs or taking into account the special multiplicative structure of the coefficients of the objective function. Another remarkable direction in studying the approximability of binary quadratic programming is to restrict the objective function to certain special types, in particular, those having *low-rank*¹ [KN07]. Intuitively, a function of low-rank can be expressed as a combination of a (fixed) number of linear functions. Allemand et al. [AFLS01] study the unconstrained 0-1 quadratic maximization problem and propose a polynomial-time algorithm when the symmetric matrix describing the coefficients of the quadratic objective function is positive semidefinite and has fixed rank². Kern and Woeginger [KW07] and Goyal et al. [GKR11] give an FPTAS for minimizing the product of two non-negative linear functions when the convex hull of feasible integer solutions is given in terms of linear inequalities. Mittal and Schulz [MS13a, MS13b] and Goyal and Ravi [GR13] extend this result to the more general class of low-rank functions.

In this paper, we consider a class of 0-1 quadratic programming, where a (nonnegative) linear or submodular objective function has to be minimized or maximized subject to a system of quadratic inequalities involving nonnegative semidefinite matrices with low *completely positive ranks*. Intuitively, the completely positive rank (cp-rank) of a semidefinite matrix Q is the smallest positive integer r such that Q can be decomposed into a sum of r non-negative rank-one matrices. As a consequence, every quadratic constraint involving such matrices can be written as the sum of r squares of (non-negative) linear functions which we call *cp-decomposition*. Hence, our problem can be seen as a generalization of the submodular maximization with (linear) knapsack constraints which has been studied extensively in the literature [Svi04, KST09, LMNS09]. Herein, we would like to emphasize that the cp-decomposition plays an important role not only in binary quadratic programming [Bur09], but also in many other areas such as block designs in combinatorial analysis [Hal86], data mining [ZS05, HXZ⁺11] and economics [VHVD08]. Several attempts have been devoted in the last decades to settling the complexity of determining if a given semidefinite positive matrix has finite cp-rank. Surprisingly, however, this has been settled only very recently in [DG14], where it was shown that this problem is NP-hard. In this paper we restrict our attention to the case of low cp-rank and study the polynomial-time algorithm for finding a cp-decomposition of a matrix whose cp-rank is fixed.

¹A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is said to be of rank k , if there exists a continuous function $g : \mathbb{R}^k \rightarrow \mathbb{R}$ and linearly independent vectors $a_1, \dots, a_k \in \mathbb{R}^n$ such that $f(x) = g(a_1^T x, \dots, a_k^T x)$, (see [KN07]).

²Note that in this case the objective function can be written as a sum of the squares of not more than k linear functions, where k is the rank of the coefficient matrix.

1.1 Our Contribution.

We summarize our contributions in this paper as follows:

- We give an $n^{O(r^2)}$ deterministic algorithm that given a positive semidefinite matrix Q of size n decides if a nonnegative factorization of inner dimension at most r exists. Furthermore, a rational approximation to the solution to within an additive accuracy of δ can be computed in time $\text{poly}(\mathcal{L}, n^{O(r^2)}, \log \frac{1}{\delta})$, where \mathcal{L} is the bit length of the input.
- We present a PTAS for maximizing a linear function subject to a fixed number of quadratic packing constraints involving positive semidefinite matrices with low cp-ranks. We provide an $(1 - \epsilon)(\frac{1}{4} - \epsilon)$ -approximation algorithm for the problem of maximizing a submodular function, for any $\epsilon > 0$. For the case when the submodular function is monotone, the approximation factor is $(1 - \epsilon)(1 - \frac{1}{e} - \epsilon)$, for any $\epsilon > 0$. Our results generalize the ones of Sviridenko [Svi04] and of Kulik et al. [KST09].
- We extend our PTAS result to a wider class of non-linear objective functions including quadratic functions, multiplicative functions, and sum-of-ratio functions.
- We give a quasi-PTAS for minimizing a linear function subject to one quadratic covering constraint involving a positive semidefinite matrix with low cp-rank.

1.2 Previous Work.

The problem of maximizing a nonnegative submodular function under knapsack constraints has been studied extensively in the literature. Sviridenko [Svi04] studied the problem for monotone submodular functions and gave a $(1 - 1/e)$ -approximation for the case of one knapsack constraint. Kulik et al. [KST09] considered the problem with any constant number of knapsack constraints, and gave a $(1 - 1/e - \epsilon)$ -approximation, for any $\epsilon > 0$. Lee et al. [LMNS09] investigated the problem of maximizing a general submodular function under a constant number of knapsack constraints, and presented algorithms that achieve approximation ratio of $1/5 - \epsilon$. A better approximation factor of $1/4 - \epsilon$ was obtained in [KST09]. In this paper, we will extend these results to the case with *quadratic* knapsack constraints having low cp-rank.

Chau et al. [CEK14] considered the problem CKP with linear objective function and gave a PTAS, which settles the complexity of the problem given the strong NP-hardness in [YC13]. As CKP is a special case of the problem of maximizing a linear function subject to a single quadratic inequality whose constraint matrix has cp-rank 2, our second result above can be thought of as a generalization of the PTAS in [CEK14] in several directions: we allow the cp-rank to be any constant, we consider any constant number of inequalities, and consider more general objective functions. In fact, our result for submodular functions is based essentially on the same geometric idea used in [CEK14].

2 Completely Positive Rank Factorizations

A positive semi-definite matrix $Q \succeq 0$ is said to have *completely positive rank* r , denoted $\text{cp-rank}(Q) = r$, if r is the smallest number of non-negative rank-one matrices into which the matrix Q can be decomposed additively, i.e., $Q = \sum_{j=1}^r q^j (q^j)^T$, where $q^j \in \mathbb{R}_+^n$ are non-negative vectors. If no such r exists then $\text{cp-rank}(Q) = +\infty$. A notion related to the cp-rank is the *nonnegative rank* of a nonnegative (not necessarily symmetric) matrix: a positive integer r is called the nonnegative rank of a matrix $P \in \mathbb{R}_+^{n \times m}$ if and only if it is the smallest number such that there exist two matrices $U \in \mathbb{R}_+^{n \times r}, V \in \mathbb{R}_+^{r \times m}$ such that $P = UV$. It is worth noting that, in general, the nonnegative rank and the cp-rank can be different from each other. In fact, for a completely positive matrix Q , it is known that the nonnegative rank is always less than or equal to the cp-rank. We refer to the textbook [BSM03] and the references therein for more discussion about the cp-rank. The complexity of deciding the existence of such a cp-decomposition for a given matrix remained open for a long time. Very recently, Dickinson and Gijben [DG14] proved that this problem is NP-hard but

left open the question whether or not it belongs to NP. The similar hardness result was also proved for the case of nonnegative rank by Vavasis [Vav09]. Therefore, it is natural to pay attention to special cases in which the factorization problem can be solved efficiently. Arora et al. [AGKM12] and Moitra [Moi13] proposed exact polynomial-time algorithms for computing a nonnegative factorization when the nonnegative rank of the input matrix is constant. Their basic idea is to transform the factorization problem into a problem of finding a solution of a system of polynomial equations which is known to be solved by an algorithm (e.g., [Ren92, BPR96]) whose running-time is polynomial in the number of equations but is exponential in the number of variables. Based on this idea, Arora et al. [AGKM12] gave a transformation to a system with nm equations and $2r^2 2^r$ variables. As a result, checking if the nonnegative rank of P is at most r can be done in $O((nm)^{2r^2 2^r})$ time. This has then been improved by Moitra [Moi13] who gave a method to exponentially reduce the number of variables to $2r^2$, thus yielding a faster algorithm with running-time $(nm)^{O(r^2)}$. It was also shown by Arora et al. [AGKM12] that achieving an exact $(nm)^{o(r)}$ -algorithm is impossible under the Exponential Time Hypothesis [IP01]. In the following, we will show that one can obtain an exact algorithm for the cp-rank case by employing the same idea as for the nonnegative rank. Assume Q is rational and let \mathcal{L} be the maximum bit length of any element in Q . The result is stated in Theorem 2.1 below.

Theorem 2.1. *There is an $n^{O(r^2)}$ time deterministic algorithm that given a positive semi-definite matrix Q of size n produces a nonnegative factorization $Q = UU^T$ of inner dimension at most r if such a factorization exists. Furthermore, a rational approximation to the solution to within an additive accuracy of δ can be computed in time $\text{poly}(\mathcal{L}, n^{O(r^2)}, \log \frac{1}{\delta})$.*

Proof. We proceed essentially along the same lines as in the proof of Theorem 2.1 in [AGKM12] (for finding a simplicial factorization). Let Q be a nonnegative positive semi-definite matrix of size n . Assume that Q has a decomposition $Q = UU^T$, where U is a nonnegative matrix of size $n \times r$. A basic fact from linear algebra is that Q, U and U^T have the same rank: $\text{rank}(Q) = \text{rank}(U) = \text{rank}(U^T) = s \leq r$.

Fix an arbitrary basis V of the column vectors of Q , let B be an $n \times s$ matrix corresponding to this basis and let Q_V be the matrix of size $s \times n$ corresponding to the (unique) representation of Q in the basis V , that is, $BQ_V = Q$. To prove the claim of the theorem, it will suffice to prove the following.

Claim 2.2. *Q has a non-negative factorization $Q = UU^T$ of inner-dimension at most r if and only if there is an $r' \times s$ matrix H , with $r' \leq r$, satisfying the following two conditions:*

- (i) HQ_V is a nonnegative matrix,
- (ii) $(HQ_V)^T(HQ_V) = Q$.

Proof. (\Leftarrow) Suppose that the conditions (i) and (ii) are satisfied. Let $U = (HQ_V)^T$. This matrix is nonnegative and has size of $n \times r$ and thus, Q has cp-rank at most $r' \leq r$.

(\Rightarrow) Now suppose that there is a non-negative factorization $Q = UU^T$, where U is an $n \times r'$ -matrix, with $r' \leq r$. The singular value decomposition of U has the form $U = LSR^T$, where L, R are orthogonal matrices ($L^{-1} = L^T$ and $R^{-1} = R^T$) of size $n \times n$ and $r' \times r'$, respectively, and S is an $n \times r'$ diagonal matrix. The first s (non-zero) diagonal entries of S are exactly the singular values of U (corresponding to the first s (non-zero) diagonal entries of Q). We can rewrite Q in the form $Q = (LSR^T)(LSR^T)^T = (LSR^T)(RS^T L^T)$. Let $N = RS^+ L^T$ (an $r' \times n$ matrix), where S^+ is the *pseudo-inverse* of matrix S . More precisely, S^+ is an $r' \times n$ matrix such that:

$$SS^+ = \begin{pmatrix} I_{s \times s} & 0 \\ 0 & 0 \end{pmatrix}_{n \times n} \quad \text{and} \quad S^+S = \begin{pmatrix} I_{s \times s} & 0 \\ 0 & 0 \end{pmatrix}_{r' \times r'},$$

where $I_{s \times s}$ is an identity matrix of size s .

Define an $r' \times s$ matrix $H = NB$. We now prove that this matrix H satisfies the two conditions (i) and (ii). Indeed, we have:

$$\begin{aligned} HQ_V &= NBQ_V = (RS^+ L^T)BQ_V = (RS^+ L^T)Q = (RS^+ L^T)(LSR^T)(RS^T L^T) \\ &= RS^+ SS^T L^T = RS^T L^T = U^T, \end{aligned}$$

from which both (i) and (ii) follow. \blacksquare

By the claim, to check if $\text{cp-rank}(Q) \leq r$ is equivalent to determining if a system of at most nr linear inequalities and n^2 quadratic equations on at most r^2 variables is feasible. This decision problem can be solved in $n^{O(r^2)}$ time using quantifier elimination algorithms [BPR96]. Furthermore, a rational approximation to the solution to within an additive accuracy of δ can be computed in time $\text{poly}(\mathcal{L}, n^{O(r^2)}, \log \frac{1}{\delta})$; see [GV88, Ren92]. \blacksquare

Corollary 2.3. *For a real matrix M let $\|M\|_\infty := \max_{i,j} |M_{ij}|$. Given a rational positive semi-definite matrix Q of size n such that $\text{cp-rank}(Q) = r$, and $\epsilon > 0$, one can compute a rational nonnegative $n \times r$ -matrix \tilde{U} such that $\tilde{U}\tilde{U}^T \geq Q$ and $\|Q - \tilde{U}\tilde{U}^T\|_\infty \leq \epsilon$, in time $\text{poly}(\mathcal{L}, n^{O(r^2)}, \log \frac{1}{\epsilon})$.*

Proof. Let $s = \text{rank}(Q)$ and B and Q_V be the matrices corresponding to basis V of Q , as in the proof of Theorem 2.1. Then from this proof, it follows that we can discover in $n^{O(r^2)}$ time that there is an $r \times s$ -real matrix H such that conditions (i) and (ii) of claim 2.2 hold. Furthermore, in time $\text{poly}(\mathcal{L}, n^{O(r^2)}, \log \frac{1}{\delta})$, we can find a rational matrix \hat{H} , such that $\|H - \hat{H}\|_\infty \leq \delta$, for any desired accuracy $\delta > 0$. Let B' be a non-singular $s \times s$ -submatrix of B , obtained by selecting s linearly independent rows of B , and let Q' be the corresponding $s \times n$ submatrix of Q . Then $Q_V = (B')^{-1}Q'$. Let us choose

$$\delta := \min \left\{ \frac{\epsilon}{\|(B')^{-1}\|_\infty \|Q\|_\infty^{3/2} r s^2 (4 + 3s\|Q\|_\infty^{1/2})}, \frac{1}{2 \cdot \max\{s\|(B')^{-1}\|_\infty, 1\}} \right\}, \quad (1)$$

and set $\delta' := s\delta\|(B')^{-1}\|_\infty$ and $\tilde{H} := \hat{H} + \delta'E_{r \times s}B'$, where $E_{r \times s}$ is the $r \times s$ -matrix with all-ones. Define $U := (HQ_V)^T$, $\hat{U} := (\hat{H}Q_V)^T$, and $\tilde{U} := (\tilde{H}Q_V)^T$. Then, it follows that

$$\tilde{U}^T = U^T + [\delta'E_{r \times s} + (\tilde{H} - \hat{H})(B')^{-1}]Q' \geq U^T + (\delta' - s\delta\|(B')^{-1}\|_\infty)E_{r \times s}Q' = U^T,$$

where the first inequality follows from the fact that $Q' \geq 0$, and the second equality follows from our choice of δ' . Let us further note that

$$\tilde{U}\tilde{U}^T - UU^T = (Q')^T \Delta^T U^T + U \Delta Q' + (Q')^T \Delta^T \Delta Q', \quad (2)$$

where $\Delta := \delta'E_{r \times s} + (\tilde{H} - \hat{H})(B')^{-1}$. To bound $\|Q - \tilde{U}\tilde{U}^T\|_\infty$, we bound the norm of each term in (2). We first note that $\|\Delta\|_\infty \leq \delta' + \delta s\|(B')^{-1}\|_\infty = 2\delta'$, and thus

$$\|U \Delta Q'\|_\infty \leq 2\delta' \|U E_{r \times s} Q'\|_\infty \leq 2\delta' r s \|U\|_\infty \|Q\|_\infty \leq 2\delta' r s \|Q\|_\infty^{3/2}, \quad (3)$$

where the last inequality follows from the fact that $\|U\|_\infty \leq \|Q\|_\infty^{1/2}$. Now we write

$$\Delta^T \Delta = (\delta')^2 E_{s \times r} E_{r \times s} + \delta' (E_{s \times r} X + X^T E_{r \times s}) + X^T X, \quad (4)$$

where $X := (\tilde{H} - \hat{H})(B')^{-1}$. From (4) and $\|X\|_\infty \leq \delta'$ follows the inequality $\|\Delta^T \Delta\|_\infty \leq 3r(\delta')^2$, and then the inequality

$$\|(Q')^T \Delta^T \Delta Q'\|_\infty \leq 3r(\delta')^2 \|(Q')^T E_{s \times s} Q'\|_\infty \leq 3r(s\delta')^2 \|Q\|_\infty^2. \quad (5)$$

Using (3) and (5) in (2), we get

$$\|\tilde{U}\tilde{U}^T - Q\|_\infty \leq 4\delta' r s \|Q\|_\infty^{3/2} + 3r(s\delta')^2 \|Q\|_\infty^2 \leq \epsilon,$$

by our selection (1) of δ . Finally note that $\log \frac{1}{\delta} \leq \text{poly}(\mathcal{L}, r \log \frac{1}{\epsilon})$. \blacksquare

3 Approximation Algorithms for Binary Quadratic Programming

Binary programming with quadratic constraints (BQC) is the problem of maximizing (or minimizing) a function of a set of binary variables, subject to a finite set of quadratic constraints. A BQC problem takes the form:

$$\begin{array}{ll}
 \text{(PACKING-BQC)} & \max f(x) \\
 \text{subject to} & x^T Q_i x \leq C_i^2, \quad i \in [m] \\
 & x \in \{0, 1\}^n,
 \end{array} \quad (6)
 \quad \left| \quad
 \begin{array}{ll}
 \text{(COVERING-BQC)} & \min f(x) \\
 \text{subject to} & x^T Q_i x \geq C_i^2, \quad i \in [m] \\
 & x \in \{0, 1\}^n,
 \end{array} \quad (7)$$

where f is a nonnegative function, and Q_i is (without loss of generality) an $n \times n$ nonnegative symmetric matrix, for all $i \in [m] := \{1, 2, \dots, m\}$. For simplicity, we fix some useful notation. For any integer $n \geq 1$, we denote by \mathcal{CP}_n the set of completely positive matrices (i.e., with finite cp-rank), and by \mathcal{CP}_n^* the subset of \mathcal{CP}_n that includes matrices of fixed cp-rank. Furthermore, we identify a vector $x \in \{0, 1\}^n$ with a subset $S \subseteq [n]$, i.e., write $S = S(x) = \{i \in [n] \mid x_i = 1\}$. Hence, for a function f defined on the power set $2^{[n]}$, $f(x) \equiv f(S)$. We shall consider non-negative functions f with a special structure: f is *linear* if $f(x) := u^T x$ for some vector $u \in \mathbb{R}_+^n$; f is *submodular* if $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$ for all subsets $S, T \subseteq [n]$; f is *quadratic*³ if $f(x) = x^T Q x + u^T x$, for some $Q \in \mathcal{CP}_n^*$ and $u \in \mathbb{R}_+^n$; we refer to the corresponding packing (covering) problems as PACK-LIN, PACK-SUB, and PACK-QUAD (COVER-LIN, COVER-SUB, and COVER-QUAD-QC), respectively. As standard in the literature, we assume that the submodular function f is given by a value oracle that always returns the value $f(S)$ for any given set $S \subseteq [n]$.

For $\alpha > 0$, a vector $x \in \{0, 1\}^n$ (or the corresponding set $S \subseteq [n]$) is said to be α -*approximate solution* for problem PACKING-BQC (resp., COVERING-BQC), if x is a feasible solution satisfying $f(x) \geq \alpha \cdot \text{OPT}$ (resp., $f(x) \leq \alpha \cdot \text{OPT}$), where OPT is the value of an optimal solution; for simplicity, if $\alpha = 1 - \epsilon$ (resp., $1 + \epsilon$), for $\epsilon > 0$, x will be called ϵ -*optimal*.

It can be seen that the problem (BQC) is NP-hard as it includes among others the *multi-dimensional Knapsack packing and covering* problems as special cases, when we set f to be a linear function and set each matrix Q_i to be a rank-one non-negative matrix. A *fully polynomial-time approximation scheme* (FPTAS)⁴ for the knapsack problem was given in [IK75], and a PTAS for the m -dimensional knapsack problem was given in [CHW76, FC84], and these results are best possible, assuming $P \neq NP$. When f is submodular, it was shown that there exist constant factor approximation algorithms for the packing problem, subject to a constant number of knapsack constraints (e.g., [Svi04, KST09]). We extend these results as follows.

Theorem 3.1. PACK-LIN admits a PTAS when m and each $\text{cp-rank}(Q_i)$ is fixed.

Theorem 3.2. For any $\epsilon > 0$, there is a $(1 - \epsilon)\alpha$ -approximation algorithm for PACK-SUB when m and each $\text{cp-rank}(Q_i)$ is fixed, where $\alpha = (1 - \frac{1}{e} - \epsilon)$, if the objective f is monotone, and $\alpha = \frac{1}{4} - \epsilon$, otherwise.

Theorem 3.3. COVER-LIN admits a QPTAS when $m = 1$, and $\text{cp-rank}(Q)$ is fixed.

The next sections are devoted to present two methods for designing approximation algorithms for PACKING-BQC. The first method relies on the polynomial solvability of the convex relaxation, whereas the second one takes into account the geometric properties of the region of feasible solutions. The first method works for the case of linear objective function and allows also for an additive linear term in each quadratic constraint; it can also be extended to handle more general classes of objective functions such as quadratic functions with fixed cp-rank, multiplicative functions, and sums of ratios of linear functions, as well as to packing constraints

³Note that a quadratic function $f : 2^{[n]} \rightarrow \mathbb{R}_+$ is *supermodular*, that is, $f(S \cup T) + f(S \cap T) \geq f(S) + f(T)$ for all subsets $S, T \subseteq [n]$.

⁴A PTAS is an algorithm that runs in time polynomial in the input size n , for every fixed ϵ , and outputs an ϵ -optimal solution; an FPTAS is a PTAS where the running time is polynomial in $\frac{1}{\epsilon}$; a QPTAS is similar to a PTAS but the running time is quasi-polynomial (i.e., of the form $n^{\text{poly}(\log n)}$), for every fixed ϵ .

involving the ℓ_p -norm for $p \geq 1$. However, we are not able to show that this method works for submodular objective functions. The second method works for the latter case, but it does not seem to have the flexibility of the first method in handling more general objective functions and/or additional linear terms in the quadratic constraints.

For the minimization problem COVER-LIN, we use the idea of geometric partitioning combined together with a greedy covering approach to arrive at the result of Theorem 3.3; see Section 3.3. We prove Theorem 3.2 in Section 3.2 and Theorem 3.1 for the linear objective case in Section 3.1; we extend the result to the quadratic objective case in Section 4.

As a technical remark, we note that the decomposition of each Q_i as given by Corollary 2.3 involves an error term Δ_i . To simplify the presentation, we will assume first that an exact decomposition $Q_i = U_i U_i^T$ is given, then show in Section 3.4 to see how to deal with the error term.

3.1 Linear Objective Function and Convex Programming-based Method

In this section, we present a polynomial-time approximation scheme for PACKING-BQC with linear objective function. This problem has the form (6) with $f(x) = u^T x$, where $Q_i \in \mathcal{CP}_n^*$ and $u \in \mathbb{R}_+^n$. For convenience, we will sometimes write $u(S) := \sum_{k \in S} u_k$, for a set $S \subseteq [n]$. The method we use here is based on solving the convex relaxation obtained by replacing the constraint $x \in \{0, 1\}^n$ by the weaker one $x \in [0, 1]^n$. The crucial point of our method is that an upper bound on the optimal value can be computed easily by solving (approximately) the relaxed problem. The obtained solution to the convex program defines a point in a certain polytope, which is then rounded to an integral solution without losing much on the quality of the solution. The details of our method is described in Algorithm 1.

Algorithm 1 PACK-LIN-PTAS($u, \{Q_i, C_i\}_{i \in [m]}, \epsilon$)

Input: Utility vector $u \in \mathbb{R}_+^n$; matrices $Q_i \in \mathcal{CP}_n^*$, capacities $C_i \in \mathbb{R}_+$, for $i \in [m]$; and accuracy parameter ϵ

Output: An ϵ -optimal solution S to PACK-LIN

```

1:  $v \leftarrow (0, \dots, 0)$ 
2: For all  $i \in [m]$ , decompose  $Q_i$  as  $Q_i = U_i U_i^T$ , where  $U_i \in \mathbb{Q}^{n \times r_i}$   $\triangleright r_i = \text{cp-rank}(Q_i)$ 
3:  $\bar{r} \leftarrow \sum_{i=1}^m (r_i + 1)$ ;  $\lambda \leftarrow \frac{\bar{r}}{\epsilon}$ 
4: for each subset  $\mathcal{U}$  of  $[n]$  of cardinality at most  $\lambda$  do
5:    $\mathcal{V} \leftarrow \{k \in [n] \setminus \mathcal{U} \mid u_k > \min\{u_{k'} \mid k' \in \mathcal{U}\}\}$ 
6:   if  $\mathbf{1}_{\mathcal{U}}^T Q_i [\mathcal{U}; \mathcal{U}] \mathbf{1}_{\mathcal{U}} \leq C_i^2$  for all  $i \in [m]$  then  $\triangleright (\text{CP1}[\mathcal{U}])$  is feasible
7:     Let  $x^*$  be an  $\epsilon$ -optimal solution to the convex program  $(\text{CP1}[\mathcal{U}])$ 
8:      $t_i \leftarrow U_i^T [*; N] x_N^*$ ;  $t'_i \leftarrow \mathbf{1}_{\mathcal{U}}^T Q_i [\mathcal{U}; N] x_N^*$  for all  $i \in [m]$   $\triangleright t_i$  is a vector of dimension  $r_i$ 
9:     Find a BFS  $y$  of the polytope  $\mathcal{P}(\mathcal{U})$  such that  $u^T y \geq u^T x_N^*$ :
10:     $\bar{x} \leftarrow \{(\bar{x}_1, \dots, \bar{x}_n) \mid \bar{x}_k = \lfloor y_k \rfloor \text{ for } k \in [n] \setminus (\mathcal{U} \cup \mathcal{V}), x_k = 1, \text{ for } k \in \mathcal{U}, x_k = 0, \text{ for } k \in \mathcal{V}\}$ 
     $\triangleright$  rounding down  $y_k$ 
11:    if  $u^T \bar{x} > u^T v$  then
12:       $v \leftarrow \bar{x}$ 
13: return  $S(v)$ 
```

Let $\epsilon > 0$ be a fixed constant, and define $\lambda = \frac{\bar{r}}{\epsilon}$, where $\bar{r} = \sum_{i=1}^m (r_i + 1)$. The idea, extending the one in [FC84], is to try to guess λ items of highest utility in the optimal solution. This can be done by considering all possibilities for choosing a set of cardinality at most λ . We denote by $\mathcal{X} \subseteq 2^{[n]}$ the set of all such sets. Note that the size of \mathcal{X} is bounded by $O(n^\lambda)$ and thus is polynomial in the size of input for every constant λ .

For each $\mathcal{U} \in \mathcal{X}$, we define a set \mathcal{V} , which contains all items that are *not* in the optimal solution, given that \mathcal{U} is a subset of the optimal solution (these are the items $k \in [n] \setminus \mathcal{U}$, with $u_k > u_{k'}$ for some $u_{k'} \in \mathcal{U}$). Let $N := [n] \setminus (\mathcal{U} \cup \mathcal{V})$. Using polynomial-time algorithms for convex programming (see, e.g., [NT08]), we

can find an ϵ -optimal solution⁵ to the convex relaxation (CP1[\mathcal{U}]) defined as follows:

$$\begin{aligned} \text{(CP1[}\mathcal{U}\text{])} \quad & \max u^T x \\ \text{subject to} \quad & x^T Q_i x \leq C_i^2, \quad i \in [m], \\ & x_k = 1 \text{ for } k \in \mathcal{U}; x_k = 0 \text{ for } k \in \mathcal{V}, \\ & x_k \in [0, 1] \text{ for } k \in N. \end{aligned}$$

We then define a polytope $\mathcal{P}(\mathcal{U}) \subseteq [0, 1]^N$ by replacing each constraint $x^T Q_i x \leq C_i^2$ by the two constraints $U_i^T[*; N]y \leq t_i$ and $\mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; N]y \leq t'_i$, for all $i \in [m]$, where $t_i := U_i^T[*; N]x_N^*$, $t'_i := \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; N]x_N^*$; here, $x_{\mathcal{U}}$ denotes the restriction of the vector $x \in [0, 1]^n$ to the set of components indexed by \mathcal{U} , and $Q_i[\mathcal{U}; N]$ denotes the restriction of the matrix Q_i to the columns and rows indexed by the sets \mathcal{U} and N , respectively; similarly, $U_i^T[*; N]$ means the restriction of U_i^T to the set of columns defined by N :

$$\mathcal{P}(\mathcal{U}) := \{y \in [0, 1]^N \mid U_i^T[*; N]y \leq t_i, \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; N]y \leq t'_i, \text{ for } i \in [m]\}.$$

We can find a basic feasible solution (BFS) y in this polytope such that $u^T y \geq u^T x_N^*$, by solving at most n linear systems (see standard texts on Linear Programming, e.g., [Sch86]). Next, an integral solution \bar{x} is obtained from y by dropping all fractional components to 0 and setting $x_k \in \{0, 1\}$ according to the assumption $k \in \mathcal{U} \cup \mathcal{V}$. For each set $\mathcal{U} \in \mathcal{X}$ such that (CP1[\mathcal{U}]) is feasible, the algorithm outputs an integral solution \bar{x} . Let v be the solution of maximum value amongst all such solutions \bar{x} . The algorithm produces a set $S(v)$ of items that corresponds to v .

Lemma 3.4. *Assume $Q_i \in \mathcal{CP}_n^*$ for all $i \in [m]$ and $m = O(1)$. Then, for any fixed $\epsilon > 0$, Algorithm 1 runs in polynomial time and produces an 2ϵ -optimal solution to the input instance.*

Proof. From the argument above, it can be easily seen that the running time of the Algorithm 1 is polynomial in size of the input, for any fixed constant ϵ . We now argue that the solution S returned by the algorithm is 2ϵ -optimal. Indeed, let S^* be the optimal solution to PACK-LIN of utility $u(S^*) = \text{OPT}$. If $\lambda \geq |S^*|$, then S is an exact optimum solution to the PACK-LIN instance. Suppose that $\lambda < |S^*|$. Let \mathcal{U} be the set of highest-utility λ items in S^* . Then, the feasibility of S^* for (CP1[\mathcal{U}]) guarantees that $\mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; \mathcal{U}] \mathbf{1}_{\mathcal{U}} \leq C_i^2$, and hence an ϵ -optimal solution x^* for (CP1[\mathcal{U}]) is found in step 7 of the algorithm. Assume w.l.o.g. that $\mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; \mathcal{U}] \mathbf{1}_{\mathcal{U}} \leq C_i^2$. From the feasibility of x^* for (CP1[\mathcal{U}]) and the factorization of Q_i , it follows that, for all $i \in [m]$,

$$(x_N^*)^T U_i[N; *] U_i^T[*; N] x_N^* + 2 \cdot \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; N] x_N^* \leq C_i^2 - \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; \mathcal{U}] \mathbf{1}_{\mathcal{U}},$$

implying that

$$t_i^2 + 2t'_i \leq C_i^2 - \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; \mathcal{U}] \mathbf{1}_{\mathcal{U}}. \quad (8)$$

The definition of t_i, t'_i implies that $x_N^* \in \mathcal{P}(\mathcal{U})$. Thus, there is a BFS y of $\mathcal{P}(\mathcal{U})$ such that $u^T y \geq u^T x_N^*$. Let $\bar{x} \in \{0, 1\}^n$ be the rounded solution obtained from y in step 10. Since any BFS of $\mathcal{P}(\mathcal{U})$ has at most \bar{r} fractional components and $u_k \leq \min_{k' \in \mathcal{U}} u_{k'}$ for all $k \in N$, it follows that

$$\begin{aligned} u^T \bar{x} &= u^T \bar{x}_N + u^T \mathbf{1}_{\mathcal{U}} \geq u^T y - \frac{\bar{r} \cdot u^T \mathbf{1}_{\mathcal{U}}}{|\mathcal{U}|} + u^T \mathbf{1}_{\mathcal{U}} \\ &\geq u^T x_N^* + (1 - \frac{\bar{r}}{\lambda}) u^T \mathbf{1}_{\mathcal{U}} \geq (1 - \epsilon) u^T x^* \geq (1 - \epsilon)^2 \text{OPT}. \end{aligned}$$

It remains to argue that \bar{x} is feasible for PACK-LIN. Since $\bar{x}_N \in \mathcal{P}(\mathcal{U})$, we have

$$\begin{aligned} \bar{x}^T Q_i \bar{x} &= \bar{x}_N^T U_i[N; *] U_i^T[*; N] \bar{x}_N + 2 \cdot \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; N] \bar{x}_N + \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; \mathcal{U}] \mathbf{1}_{\mathcal{U}} \\ &\leq t_i^2 + 2t'_i + \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; \mathcal{U}] \mathbf{1}_{\mathcal{U}} \leq C_i^2, \end{aligned}$$

⁵In fact, such algorithms can find a feasible solution x^* to the convex relaxation such that $u^T x^* \geq \text{opt} - \delta$, in time polynomial in the input size (including the bit complexity) and $\log \frac{1}{\delta}$, where opt is the value of the fractional optimal solution. We may assume that $\mathbf{1}_k^T Q_i \mathbf{1}_k \leq C_i^2$, for all $i \in [m]$, where $\mathbf{1}_k$ is the k th unit vector; otherwise item k can be removed from the (PACK-LIN) problem. This implies that $\text{opt} \geq \text{OPT} \geq \bar{u} := \max_k u_k$. Now setting δ to $\epsilon \cdot \bar{u}$ assures that $u^T x^* \geq (1 - \epsilon) \text{opt}$.

where the last inequality follows from (8). This completes the proof. \blacksquare

Remark 3.5. Note that the result presented in this section can be extended easily to the following problem:

$$\begin{aligned} & \max u^T x \\ \text{subject to} \quad & x^T Q_i x + q_i^T x \leq C_i^2, \quad i \in [m] \\ & Ax \leq b, \\ & x \in \{0, 1\}^n, \end{aligned}$$

where $Q_i \in \mathcal{CP}_n^*$, $q_i \in \mathbb{R}_+^n$, for all $i \in [m]$; $A \in \mathbb{R}_+^{d \times n}$, $b \in \mathbb{R}_+^d$; m, d are constants.

3.2 Submodular Objective Function and Geometry-based Method

In this section, we present a constant factor approximation algorithm for PACK-SUB. This problem has the form (6) where $f(x)$ is a nonnegative submodular function, $Q_i \in \mathcal{CP}_n^*$, and m is constant.

If one attempts to use the technique of the previous section, (s)he runs into the difficulty of how to relax the objective function; the two well-known relaxations (see, e.g., [Dug09] for a recent survey) are: the *Lovász extension* which is known to be *convex* [GLS88], and the *multi-linear extension* [Von08] which is in general *neither convex nor concave*. While it is not known how to solve the maximization problems for the relaxations corresponding to these two types of functions over a polytope in polynomial time, it is known [Von08, VCZ11] how to approximate within a constant factor the maximum of the multi-linear extension over a packing polytope. It is not clear if this result can be extended to the case when the feasible region is described by convex quadratic constraints of the form $x^T Q_i x \leq C_i^2$, where $Q_i \in \mathcal{CP}_n^*$. While this remains an interesting open question, we will rely here, instead, on a geometric-based approach that uses the results in [VCZ11, KST09] as a black-box. The key idea is to make use of the geometry of the problem to reduce it into a multi-dimensional knapsack problem, which can be solved using enumeration and dynamic programming for the linear objective case, or LP-rounding for the submodular case.

Let $I = (f, \{Q_i, C_i\}_{i \in [m]})$ be an instance of the problem PACK-SUB, and $\epsilon > 0$ be a fixed constant. We will construct a polynomial-time algorithm which produces a constant factor-approximate solution to the instance I . Write $Q_i = U_i U_i^T$, for some $U_i \in \mathbb{Q}_+^{n \times r_i}$, where $r_i = \text{cp-rank}(Q_i)$. For $k \in [n]$, define the vector $q_k^i \in \mathbb{Q}_+^{r_i}$ to be the k th column of U_i^T . Since $\bar{r} \triangleq \max_{i \in [m]} r_i$ is bounded by a constant, we may assume without loss of generality that $\epsilon < \frac{1}{4\bar{r}}$. For $r \in \mathbb{Z}_+$ and $C \in \mathbb{R}_+$, define $\mathcal{B}(r, C) \triangleq \{\nu \in \mathbb{R}_+^r : \|\nu\|_2 \leq C\}$ to be the non-negative sector of the ball in \mathbb{R}^r of radius C centered at the origin. Then problem PACK-SUB amounts to finding an $S \subseteq [n]$, maximizing $f(S)$, such that $\sum_{k \in S} q_k^i \in \mathcal{B}(r_i, C_i)$ for all $i \in [m]$.

Given a feasible set $T \subseteq [n]$ (that is, $\|d_T\|_2 \leq C_i$ for all $i \in [m]$), we define \mathcal{R}_T^i as the conic region bounded as the following:

$$\mathcal{R}_T^i \triangleq \left\{ \nu \in \mathbb{R}_+^{r_i} : \|\nu\|_2 \leq C_i^2, \nu \geq q_T^i \right\}, \quad (9)$$

where $q_T^i \triangleq \sum_{k \in T} q_k^i$. Write $q_k^i \triangleq (q_k^{ij} : j \in [r_i]) \in \mathbb{R}^{r_i}$. Given \mathcal{R}_T^i , we define an r_i -dimensional grid in the region \mathcal{R}_T^i by interlacing equidistant $(r_i - 1)$ -dimensional parallel hyperplanes with inter-separation $\frac{\epsilon}{2r_i} w_T^{ij}$, for each $j \in [r_i]$, and with the j th principal axis as their normal, where

$$w_T^{ij} \triangleq \sqrt{C_i^2 - \sum_{j' \neq j} (q_T^{ij'})^2} - q_T^{ij} \quad (10)$$

is the distance along the j th axis from q_T^i to the boundary of the ball⁶. For $j \in [r_i]$, let H^{ij} be such hyperplane that is furthest from the origin, perpendicular to the j th principal direction, and let H_+^{ij} be the half-space

⁶To maintain *finite precision*, we need to *round down* w_T^{ij} in (10) to the nearest integer multiple of $\frac{1}{D}$, where D is the common denominator of all the rational number q_k^{ij} ; this should be done also for any computations involving square roots in the sequel. This introduces an error that can be dealt with in a similar way as we deal with the error caused by the Δ_i 's. To keep the presentation simple, we will assume infinite precision.

defined by H^{ij} that includes the origin. This grid defines at most $r_i(\frac{2r_i}{\epsilon})^{r_i-1}$ lines that intersect the spherical boundary of region \mathcal{R}_T^i at a set of points $P_T^i(\epsilon)$. Let us call a cell (that, is a hypercube) of the grid *boundary* if it overlaps with the surface of the ball $\mathcal{B}(r_i, C_i)$ and denote by \mathcal{B}_T^i the set of all such boundary cells. The convex hull of the set of points $P_T^i(\epsilon)$ defines a polytope \mathcal{Q}_T^i with the following properties; the first two are easily verifiable; the last one follows from the *upper bound Theorem* [McM70] and the algorithms for convex hull computation in fixed dimension (e.g. [Cha93]):

- (I) All points in $P_T^i(\epsilon)$ are vertices of \mathcal{Q}_T^i .
- (II) Let \mathcal{H}_T^i be the set of half-spaces including the origin and defined by the facets of \mathcal{Q}_T^i . Then for each $H_+ \in \mathcal{H}_T^i$, the facet defining H_+ is the convex hull of a set of vertices from $P_T^i(\epsilon)$ that are contained completely inside a boundary cell, and its normal is a *nonnegative* vector.
- (III) The number of facets of \mathcal{Q}_T^i is at most $(\frac{2r_i}{\epsilon})^{r_i^2/2}$ and they can be found in time $O((\frac{2r_i}{\epsilon})^{r_i-1}(r_i^2 \log \frac{r_i}{\epsilon} + (\frac{2r_i}{\epsilon})^{((r_i-1)^2+1)/2}))$.

Let $\mathcal{P}_T^i(\epsilon)$ be the polytope defined by the intersection

$$\left(\bigcap_{H_+ \in \mathcal{H}_T^i} H_+ \right) \cap \left(\bigcap_{j \in [r_i]} H_+^{ij} \right) \cap \{\nu \in \mathbb{R}^{r_i} : \nu \geq 0\},$$

and denote by $m_T^i(\epsilon)$ the number of facets of $\mathcal{P}_T^i(\epsilon)$. By construction, $m_T^i(\epsilon) \leq (\frac{2r_i}{\epsilon})^{r_i^2/2} + 2r_i$.

Consider a collection of feasible subsets $\mathcal{T} \triangleq \{T^1, \dots, T^m\}$, $T^i \subseteq [n]$, to PACK-SUB. We define the following approximate version of problem PACK-SUB, which is obtained by approximating the ball sectors $\mathcal{B}(r_i, C_i)$ by inscribed polytopes.

$$\begin{aligned} & \text{(PQP}_{\mathcal{T}}) \quad \max_{x \in \{0,1\}^n} f(x) \\ & \text{subject to} \quad \sum_{k \in [n]} q_k^i x_k \in \mathcal{P}_{T^i}^i(\epsilon), \quad \text{for all } i \in [m]. \end{aligned}$$

Given two vectors $\mu, \nu \in \mathbb{R}^{r_i}$, we denote the (vector) projection of μ on ν by $\text{Pj}_{\nu}(\mu) \triangleq \frac{\nu}{\|\nu\|_2^2} \mu^T \nu$. Given the polytope $\mathcal{P}_T^i(\epsilon)$, we define a set of $m_T^i(\epsilon)$ vectors $\{\sigma_{\ell}^{T^i, i}\}$ in \mathbb{R}^{r_i} , each of which is perpendicular to a facet of $\mathcal{P}_T^i(\epsilon)$, starting at the origin, and ending at the facet.

For a collection of feasible subsets $\mathcal{T} \triangleq \{T^1, \dots, T^m\}$ to PACK-SUB, we define a submodular function maximization problem subject to \bar{m} knapsack constraints based on $\{\sigma_{\ell}^{T^i, i}\}$, where $\bar{m} \triangleq \sum_{i \in [m]} m_T^i(\epsilon)$:

$$\begin{aligned} & (\bar{m}\text{DKS-SUB}\{\sigma_{\ell}^{T^i, i}\}) \quad \max_{x \in \{0,1\}^n} f(x) \\ & \text{subject to} \quad \sum_{k \in [n]} \|\text{Pj}_{\sigma_{\ell}^{T^i, i}}(q_k^{ij})\|_2 x_k \leq \|\sigma_{\ell}^{T^i, i}\|_2, \quad \text{for all } \ell = 1, \dots, m_T^i(\epsilon). \end{aligned}$$

The following lemma follows straightforwardly from the convexity of the polytopes $\mathcal{P}_T^i(\epsilon)$.

Lemma 3.6. *Given a collection of feasible solutions $\mathcal{T} \triangleq \{T^1, \dots, T^m\}$ to PACK-SUB, problems $\text{PQP}_{\mathcal{T}}$ and $\bar{m}\text{DKS-SUB}\{\sigma_{\ell}^{T^i, i}\}$ are equivalent.*

Our approximation algorithm for PACK-SUB is described in Algorithm 2 below, which enumerates every collection of sets $\mathcal{T} = \{T^1, \dots, T^m\}$ s.t. $|T^i| \leq \frac{1}{\epsilon}$, then finds a near optimal solution for $\text{PQP}_{\mathcal{T}}$ using an approximation algorithm for $\bar{m}\text{DKS-SUB}$ (e.g., the algorithm in [KST09]).

Algorithm 2 PACK-SUB-APPROX($f, \{Q_i, C_i\}_{i \in [m]}, \epsilon, \alpha(\epsilon)$)

Input: A submodular function $f : \{0, 1\}^n \rightarrow \mathbb{R}_+$; matrices $Q_i \in \mathcal{CP}_n^*$, capacities $C_i \in \mathbb{R}_+$, for $i \in [m]$; accuracy parameter ϵ ; and an $\alpha(\epsilon)$ -approximation algorithm for \bar{m} DKS-SUB with $\bar{m} = O(1)$

Output: $(1 - \epsilon)^m \alpha(\epsilon)$ -approximate solution \hat{S} to PACK-SUB

- 1: For all $i \in [m]$, decompose Q_i as $Q_i = U_i U_i^T$, where $U_i \in \mathbb{Q}_+^{n \times r_i}$ $\triangleright r_i = \text{cp-rank}(Q_i)$
 - 2: $\hat{S} \leftarrow \emptyset$
 - 3: **for** each collection of sets $\mathcal{T} = \{T^1, \dots, T^m\}$ s.t. $T^i \subseteq [n]$ and $|T^i| \leq \frac{1}{\epsilon}$ **do**
 - 4: Set $q_{T^i}^i \leftarrow \sum_{k \in T^i} q_k^i$, and define the corresponding vectors $\{\sigma_{\ell}^{T^i, i}\}$
 - 5: Obtain an $\alpha(\epsilon)$ -approximate solution S to \bar{m} DKS-SUB $\{\sigma_{\ell}^{T^i, i}\}$
 - 6: **if** $u(\hat{S}) < u(S)$ **then**
 - 7: $\hat{S} \leftarrow S$
 - 8: **return** \hat{S}
-

Theorem 3.7. For any fixed $\epsilon > 0$, if the problem of maximizing a submodular function subject to multiple knapsack constraints can be approximated within some constant factor $\alpha(\epsilon)$, then Algorithm 2 runs in polynomial in size of the input and finds a $(1 - \epsilon)^m \alpha(\epsilon)$ -approximate solution to PACK-SUB.

Proof. It is easy to see that Algorithm 2 runs in polynomial time in the number of items n . Let S^* be an optimal solution. To establish the approximation ratio for Algorithm 2, we explicitly construct a set $S \subseteq S^*$ in Lemma 3.8 below such that $u(S) \geq (1 - \epsilon)^m u(S^*)$, and S is feasible to PQP $_{\mathcal{T}}$ for some collection $\mathcal{T} = \{T^1, \dots, T^m\}$ s.t. $T^i \subseteq [n]$ and $|T^i| \leq \frac{1}{\epsilon}$. By Lemma 3.6, invoking the $\alpha(\epsilon)$ -approximation algorithm for \bar{m} DKS-SUB $\{\sigma_{\ell}^{T^i, i}\}$ gives an $\alpha(\epsilon)$ -approximation \hat{S} to PQP $_{\mathcal{T}}$. Then it follows that

$$u(\hat{S}) \geq \alpha(\epsilon) u(S) \geq (1 - \epsilon)^m \alpha(\epsilon) \text{OPT}. \quad (11)$$

The proof is thus completed by Lemmas 3.8-3.10 below. ■

Lemma 3.8. Consider a feasible solution S^* to PACK-SUB. Algorithm 3 returns a subset $S \subseteq S^*$ and a collection $\mathcal{T} = \{T^1, \dots, T^m\}$, such that (i) $T^i \subseteq S^*$ and $|T^i| \leq \frac{1}{\epsilon}$; (ii) S is a feasible solution to PQP $_{\mathcal{T}}$ and $u(S) \geq (1 - \epsilon)^m u(S^*)$.

Proof. Starting from S^* , where $\sum_{k \in S^*} q_k^1 \in \mathcal{B}(r_1, C_1)$, Algorithm 3 first finds sets $T^1, S^1 \subseteq S^*$ such that $|T^1| \leq \frac{1}{\epsilon}$, $u(S^1) \geq (1 - \epsilon) u(S^*)$, and the set of vectors in S^1 can be packed inside the polytope $\mathcal{P}_{T^1}^1(\epsilon)$.

The way for finding such sets is as follows. Let $\bar{\ell}$ and $T_{\bar{\ell}}$ be the values of ℓ and T at the end of the repeat-until loop (line 8). The algorithm first constructs a nested sequence $T_0 \subset T_1 \subset \dots \subset T_{\bar{\ell}}$, such that a vector q_k^1 is included in each iteration if it has a "large" component q_k^{1j} for some $j \in [r_1]$. The iteration proceeds until a sufficiently large number of vectors have been accumulated (namely, $|T_{\bar{\ell}}| \geq \frac{1}{\epsilon}$), or no vectors with large components remain. At the end of the iteration, if the condition in line 9 holds, then set $T^1 = T_{\bar{\ell}}$; otherwise, the algorithm finds a subset of S^* that belongs to $\mathcal{P}_{T_{\bar{\ell}}}^1(\epsilon)$. To do so, the set $S^* \setminus T_{\bar{\ell}}$ of remaining vectors is partitioned into at least $\frac{1}{\epsilon} - 1$ groups such that, along one of the principal axes, the total sum in each group is "large". Then some vector or group of vectors is dropped, ensuring that the set S^1 of remaining vectors can be packed inside $\mathcal{P}_{T^1}^1(\epsilon)$.

We now have to consider two cases (line 12): (i) $|T_{\bar{\ell}}| \geq \frac{1}{\epsilon}$, or (ii) $S_{\bar{\ell}} = \emptyset$. For case (i), the algorithm proceeds to line12 - combining the vectors in $S \setminus T_{\bar{\ell}}$ into one group V_1 . For case (ii), the set $S \setminus T_{\bar{\ell}}$ can be partitioned into at least $\frac{1}{\epsilon} - 1$ groups V_1, \dots, V_h due to Lemma 3.9, where each group V_p , $p \in [h]$, has a large total component along one of principal axes (precisely, greater than $\frac{\epsilon}{2r_1} w_{T_{\bar{\ell}}}^{1j}$ for some $j \in [r_1]$). We define S^1 by deleting some vector or group of vectors from S (lines 22 and 25). Hence, in both cases,

$S^* = \bigcup_{k \in T_\ell} \{k\} \cup \bigcup_{p \in [h]} V_p$ induces a partition of S^* into $|T_\ell| + h \geq \frac{1}{\epsilon}$ subsets. By Lemma 3.10 below, we have

$$f(S^1) \geq (1 - \frac{1}{|T_\ell| + h})f(S^*) \geq (1 - \epsilon)f(S^*).$$

Note that removing any one vector $k \in T_\ell$ or group V_p from S^* will insure that the resulting set S^1 satisfies $q_{S^1}^1 \in \mathcal{P}_{T_\ell}^1(\epsilon)$, since the lengths $w_{T_\ell}^{1j}$ are monotone decreasing in ℓ , for all $j \in [r_1]$, and the dropped vector or group of vectors has total length at least $\frac{\epsilon}{2r_i} w_T^{1j}$ along one of the principal directions $j \in [r_1]$; on the other hand, the boundary cell that contains the original point $q_{S^*}^1$ has length at most $\frac{\epsilon}{2r_i} w_T^{1j}$ along the j th direction. It follows from property (II) stated above that reducing j th component of $q_{S^*}^1$ by $\frac{\epsilon}{2r_i} w_T^{1j}$ guarantees that the resulting vector lies in the polytope $\mathcal{P}_{T_\ell}^1(\epsilon)$.

Now starting from S^1 which satisfies $\sum_{k \in S^1} q_k^2 \in \mathcal{B}(r_2, C_2)$, we find sets $T^2, S^2 \subseteq S^1$ such that $|T^2| \leq \frac{1}{\epsilon}$, $f(S^2) \geq (1 - \epsilon)f(S^1) \geq (1 - \epsilon)^2 f(S^*)$ and the set of demands in S^2 can be packed inside the polytope $\mathcal{P}_{T^2}^2(\epsilon)$, and so on. Finally, we find the set $\{T^1, \dots, T^m, S'\}$ that satisfies the lemma. The details are given in Algorithm 3. \blacksquare

Algorithm 3 QC-CONSTRUCT($f, \{q_k^i\}_{k \in S^*}, \{C_i\}_{i \in [m]}, \epsilon$)

Input: A submodular function $f : \{0, 1\}^n \rightarrow \mathbb{R}_+$; vectors $\{q_k^i\}_{k \in S^*, i \in [m]}$; capacities $\{C_i\}_{i \in [m]}$; accuracy parameter ϵ

Output: A pair $(\mathcal{T} = \{T^1, \dots, T^m\}, S)$ of a collection of sets $T^i \subseteq S^*$ and a subset $S \subseteq S^*$

```

1:  $S \leftarrow S^*$ 
2: for  $i = 1, \dots, m$  do
3:    $\ell \leftarrow 0; T \leftarrow \emptyset$ 
4:   repeat  $\triangleright$  Find a subset of large vectors  $T$  w.r.t. to the  $i$ th constraint
5:      $\ell \leftarrow \ell + 1$ 
6:      $S_\ell \leftarrow \{k \in S \setminus T \mid \exists j \in [r_i] : q_k^{ij} > \frac{\epsilon}{2r_i} w_T^{ij}\}$ , where  $w_T^{ij}$  is given by (10)
7:      $T \leftarrow T \cup S_\ell$ 
8:   until  $|T| \geq \frac{1}{\epsilon}$  or  $S_\ell = \emptyset$  or  $S \setminus T = \emptyset$ 
9:   if  $q_S^i \in \mathcal{P}_{T_\ell}^i(\epsilon)$  then
10:     $T^i \leftarrow T$ 
11:   else  $\triangleright$  Find a subset of  $S$  that belongs to  $\mathcal{P}_{T_\ell}^i(\epsilon)$ 
12:     if  $|T| \geq \frac{1}{\epsilon}$  then
13:        $T \leftarrow$  the set of the first  $\frac{1}{\epsilon}$  elements added to  $T$ 
14:        $h \leftarrow 1; V_1 \leftarrow S \setminus T$ 
15:     else
16:        $T^i \leftarrow T$ 
17:       Find a partition  $V_1, \dots, V_h$  of  $S \setminus T$  s.t.  $\exists j \in [r_i] : \sum_{k \in V_p} q_k^{ij} \geq \frac{\epsilon}{2r_i} w_T^{ij} \quad \forall p \in [h]$ 
18:        $K \leftarrow \{k \in T \mid f(S \setminus \{k\}) \geq (1 - \epsilon)f(S)\}$ 
19:        $P \leftarrow \{p \in [h] \mid f(S \setminus V_p) \geq (1 - \epsilon)f(S)\}$ 
20:       if  $K \neq \emptyset$  then
21:         Pick arbitrary  $\hat{k} \in K$ 
22:          $S \leftarrow S \setminus \{\hat{k}\}$ 
23:       else
24:         Pick arbitrary  $\hat{p} \in P$ 
25:          $S \leftarrow S \setminus V_{\hat{p}}$ 
26: return  $(\{T^1, \dots, T^m\}, S)$ 

```

Lemma 3.9. Consider sets $T \subseteq S \subseteq [n]$ such that (C1) $\sum_{k \in S} q_k^i \in \mathcal{B}(r_i, C_i) \setminus \mathcal{P}_T^i(\epsilon)$; (C2) $q_k^{ij} \leq \frac{\epsilon}{2r_i} w_T^{ij}$, for all $j \in [r_i]$ and $k \in S \setminus T$. Then there exist $j \in [r_i]$, $h \in [\frac{1}{\epsilon} - 1, \frac{2r_i}{\epsilon}]$, and a partition $\{V_1, \dots, V_h\}$ of $S \setminus T$ such that $\sum_{k \in V_s} q_k^{ij} \geq \frac{\epsilon}{2r_i} w_T^{ij}$ for all $s \in [h]$.

Proof. We define $\eta \triangleq \sum_{k \in T} q_k^i$ and $\kappa \triangleq \sum_{k \in S \setminus T} q_k^i$. Since $\eta + \kappa = \sum_{k \in S} q_k^i \in \mathcal{B}(r_i, C_i) \setminus \mathcal{P}_T^i(\epsilon)$, we claim that there is a $j \in [r_i]$ such that $\kappa^j > \frac{w_T^{ij}}{r_i}$. Indeed, consider the $(r_i - 1)$ -dimensional simplex whose j th vertex, for $j \in [r_i]$, is $(\eta^1, \dots, \eta^{j-1}, \eta^j + w_T^{ij}, \eta^{j+1}, \dots, \eta^{r_i})$. Then the point $\xi \in \mathbb{R}^{r_i}$, defined by $\xi^j = \eta^j + \frac{w_T^{ij}}{r_i}$ for $j \in [r_i]$, lies on the simplex, implying that the whole box $\mathcal{C} := \{\nu \in \mathbb{R}^{r_i} : \eta \leq \nu \leq \xi\}$ lies on the same side, including η , of the hyperplane H defined by the simplex. On the other hand, any facet of $\mathcal{P}_T^i(\epsilon)$ is defined by the convex hull of a set of points on the ball, which lie on the other side of H . Consequently, the point $\eta + \kappa$ strictly lies on this other side, and hence outside the box \mathcal{C} , implying the claim.

Let us fix $j \in [r_i]$ as in the claim, and pack consecutive vectors q_k^i , $k \in S \setminus T$, into batches such that the sum in each batch has the j th component of length in the interval $(\frac{\epsilon}{2r_i} w_T^{ij}, \frac{\epsilon}{r_i} w_T^{ij}]$. More precisely, we fix an order on $S \setminus T := \{1, \dots, \ell\}$, and find indices $1 = k_1 < k_2 < \dots < k_{h'} < k_{h'+1} = \ell + 1$ such that

$$\sum_{k=k_l}^{k_{l+1}-1} q_k^{ij} \leq \frac{\epsilon}{r_i} w_T^{ij}, \text{ for } l = 1, \dots, h' \quad (12)$$

$$\text{and } \sum_{k=k_l}^{k_{l+1}} q_k^{ij} > \frac{\epsilon}{r_i} w_T^{ij} \text{ for } l = 1, \dots, h' - 1; \quad (13)$$

the existence of such indices is guaranteed by (C2). It follows from (13) that $\sum_{k=k_l}^{k_{l+1}-1} q_k^{ij} > \frac{\epsilon}{2r_i} w_T^{ij}$ for $l = 1, \dots, h' - 1$, since $q_{k_{l+1}}^{ij} \leq \frac{\epsilon}{2r_i} w_T^{ij}$. It also follows that $\frac{1}{\epsilon} \leq h' < \frac{2r_i}{\epsilon} + 1$, since summing (12) for $\ell = 1, \dots, h'$ yields

$$\frac{\epsilon}{r_i} h' w_T^{ij} \geq \sum_{l=1}^{h'} \sum_{k=k_l}^{k_{l+1}-1} q_k^{ij} = \sum_{k=1}^{\ell} q_k^{ij} = \kappa^j \geq \frac{w_T^{ij}}{r_i}; \quad (14)$$

the last inequality follows from our assumption. Similarly, summing (13) for $\ell = 1, \dots, h' - 1$ yields

$$(h' - 1) \frac{\epsilon}{r_i} w_T^{ij} < \sum_{l=1}^{h'-1} \sum_{k=k_l}^{k_{l+1}} q_k^{ij} \leq 2 \sum_{k=1}^{r_i} q_k^{ij} = 2\kappa^j \leq 2w_T^{ij}, \quad (15)$$

where the last inequality follows from (C1). Setting $V_l \triangleq \{k_l, k_l + 1, \dots, k_{l+1} - 1\}$, for $l = 1, \dots, h' - 2$, $V_{h'-1} = \{k_{h'-1}, \dots, r\}$, and $h \triangleq h' - 1$ will satisfy the claim of the Lemma. \blacksquare

Lemma 3.10. Let $f : 2^{[n]} \rightarrow \mathbb{R}_+$ be a nonnegative submodular function and let $S \subseteq [n]$ be a non-empty set. If $\{S_1, \dots, S_k\}$ is a partition of S into k disjoint subsets, then $f(S \setminus S_i) \geq (1 - \frac{1}{k})f(S)$ for some $i \in [k]$.

Proof. We prove the claim by contradiction. Suppose that $f(S \setminus S_i) < (1 - \frac{1}{k})f(S)$ holds for all $i \in [k]$. We claim by induction on $i = 1, 2, \dots, k$ that

$$f(S \setminus \bigcup_{j=1}^i S_j) < \left(1 - \frac{i}{k}\right) f(S), \quad (16)$$

which, when applied with $i = k$, would give the contradiction $f(\emptyset) < 0$. The claim is true for $i = 1$ by assumption. Let us assume it is true up to $i - 1$. By the submodularity of f , we have:

$$f\left((S \setminus \bigcup_{j=1}^{i-1} S_j) \cup (S \setminus S_i)\right) + f\left((S \setminus \bigcup_{j=1}^{i-1} S_j) \cap (S \setminus S_i)\right) \leq f\left(S \setminus \bigcup_{j=1}^{i-1} S_j\right) + f(S \setminus S_i)$$

implying by the induction hypothesis and the assumption that $f(S \setminus S_i) < (1 - \frac{1}{k})f(S)$ that

$$f(S) + f\left(S \setminus \bigcup_{j=1}^i S_j\right) < \left(1 - \frac{i-1}{k}\right) f(S) + \left(1 - \frac{1}{k}\right) f(S),$$

proving the claim. ■

The following results are straightforward consequences of Theorem 3.7 and the results in [KST09].

Corollary 3.11. *There is a $(1 - \epsilon)^m(\frac{1}{4} - \epsilon)$ -approximation algorithm for the PACK-SUB problem, for any $\epsilon > 0$, when m and each $r_i \triangleq \text{cp-rank}(Q_i)$ are fixed. For the case when the submodular function is monotone, the approximation factor is $(1 - \epsilon)^m(1 - \frac{1}{e} - \epsilon)$, for any $\epsilon > 0$.*

Remark 3.12. Note that a slight modification of Algorithm 2 and Algorithm 3 yields a PTAS for the PACK-LIN problem. To show that, we do some following changes. Firstly, in the step 5 of Algorithm 2, the solution S will be computed by a PTAS for the m -dimensional knapsack problem instead of using the $\alpha(\epsilon)$ -approximation algorithm for the submodular objective case. As a result, we have $u(\hat{S}) \geq (1 - \epsilon)u(S') \geq (1 - \epsilon)^{m+1}\text{OPT}$. Secondly, for Algorithm 3, we do the same steps 1-25. Then, the solution S^i will be obtained from S^{i-1} by dropping the smallest utility-demand or group of demands with large component, ensuring that the set S^i of remaining demands can be packed inside $\mathcal{P}_{T^i}^i(\epsilon)$. Hence, in the former case, $u(S^i) \geq (1 - \epsilon)u(S^{i-1})$, and in the latter case, $u(S^i) \geq (1 - \frac{1}{h})S^{i-1} \geq \frac{1-2\epsilon}{1-\epsilon}S^{i-1} \geq (1 - 2\epsilon)S^{i-1}$.

3.3 Minimization Problem – a Greedy-based Approach

In this section we consider problem COVER-LIN with one quadratic constraint. We want to minimize a linear function $f(x) = u^T x$, subject to $x^T Q x \geq C^2$ and $x \in \{0, 1\}^n$, where $Q \in \mathcal{CP}_n^*$ and $u \in \mathbb{R}_+^n$. Note that the convex programming-based method can not be applied here since the relaxed problem, which is obtained by considering $x_i \in [0, 1]$ for all $i \in [n]$, is *non-convex*, and thus we do not know if it can be solved efficiently. Furthermore, one can easily show that this programming relaxation has a bad integrality gap (of at least C^2), and thus is not a good choice for approximation. Instead, we will follow a geometric approach.

Let $I = (u, Q, C)$ be an instance of the problem COVER-LIN, and $\epsilon > 0$ be a fixed constant. We will construct a quasi-polynomial-time algorithm which produces an $(1 + \epsilon)$ -approximate solution to the instance I .

By guessing, we may assume that $B \leq \text{OPT} < (1 + \epsilon)B$, where $B := (1 + \epsilon)^i \min_k u_k$ for some $i \in \mathbb{Z}_+$ (the number of possible guesses is $O(\log_{1+\epsilon}(n \cdot \max_k u_k / \min_k u_k))$). Let $T := \{k \mid u_k \geq (1 + \epsilon)B\}$ and $V := \{k \mid u_k < \frac{\epsilon}{n} \cdot B\}$; we set $x_k = 1$ for all $k \in T$, and $x_k = 0$ for all $k \in V$, and assume therefore that we need to optimize over a set $N := [n] \setminus (T \cup V)$, for which $u_k \in [\frac{\epsilon}{n} \cdot B, (1 + \epsilon)B]$ for $k \in N$. Note that such restriction increases the cost of the solution obtained by at most $\epsilon \cdot \text{OPT}$.

As before, write $Q = UU^T$, for some $U \in \mathbb{Q}_+^{n \times r}$, where $r = \text{cp-rank}(Q)$, and for $k \in [n]$ define the vector $q_k \in \mathbb{Q}_+^r$ to be the k th column of U^T . Define the conic region \mathcal{R}_T as in (9) (with the index i dropped). Then the problem now amounts to finding $S \subseteq N$ s.t. $q_S := \sum_{k \in S} q_k$ is not in the interior of \mathcal{R}_T .

We begin by partitioning the set of vectors (indices) in N into $h := r \left(\frac{\sqrt{r}}{\epsilon} \right)^{r-1}$ space-classes N^1, \dots, N^h , with the following property: for all $s \in [h]$, there exists $\xi(s) \in \mathbb{R}_+^r$ such that for all $k \in N^s$, it holds

$$\frac{q_k^T \xi(s)}{\|q_k\|_2 \|\xi(s)\|_2} \geq 1 - \epsilon. \quad (17)$$

Condition (17) says that there is a fixed direction $\xi(s)$ such that the angle that any vector q_k , $k \in N^s$, makes with $\xi(s)$ is sufficiently small.

We will rely on the following geometric facts in our analysis of the algorithm.

Fact 3.13. *Let $a, b, \xi \in \mathbb{R}_+^r$ be such that $\frac{a^T \xi}{\|a\|_2 \|\xi\|_2} \geq 1 - \epsilon$ and $\frac{b^T \xi}{\|b\|_2 \|\xi\|_2} \geq 1 - \epsilon$. Then $\frac{(a+b)^T \xi}{\|a+b\|_2 \|\xi\|_2} \geq 1 - \epsilon$.*

Proof. Indeed,

$$\begin{aligned} \frac{(a+b)^T \xi}{\|a+b\|_2 \|\xi\|_2} &= \frac{\|a\|_2}{\|a+b\|_2} \cdot \frac{a^T \xi}{\|a\|_2 \|\xi\|_2} + \frac{\|b\|_2}{\|a+b\|_2} \cdot \frac{b^T \xi}{\|b\|_2 \|\xi\|_2} \\ &\geq \frac{\|a\|_2 + \|b\|_2}{\|a+b\|_2} (1 - \epsilon) \geq 1 - \epsilon, \end{aligned}$$

by the triangular inequality. ■

Note that Fact 3.13 implies that, for any $S \subseteq N^s$, $q_S := \sum_{k \in S} q_k$ also satisfies the condition (17).

Fact 3.14. *Let $a, b, \xi \in \mathbb{R}_+^r$ be such that $\frac{a^T \xi}{\|a\|_2 \|\xi\|_2} \geq 1 - \epsilon$ and $\frac{b^T \xi}{\|b\|_2 \|\xi\|_2} \geq 1 - \epsilon$. Then $\frac{a^T b}{\|a\|_2 \|b\|_2} \geq 1 - 5\epsilon$.*

Proof. Let $\bar{a} := \frac{a}{\|a\|_2}$, $\bar{b} := \frac{b}{\|b\|_2}$, and $\bar{\xi} := \frac{\xi}{\|\xi\|_2}$. If a, b, ξ all lie in the same subspace, then the claim follows since the angle between a and b is no more than the sum of the angles between a and ξ , and b and ξ , which is at most $\cos^{-1}((1 - \epsilon)^2 - \epsilon(2 - \epsilon)) \leq \cos^{-1}(1 - 4\epsilon)$. Otherwise, let $\bar{b} = \hat{b} + \tilde{b}$ be the orthogonal decomposition of \bar{b} with respect to the 2-dimensional subspace formed by the two vectors \bar{a} and $\bar{\xi}$, where \hat{b} is the projection of \bar{b} into this space, and \tilde{b} is the orthogonal component. Then $\frac{\hat{b}^T \bar{\xi}}{\|\hat{b}\|_2} = \frac{\bar{b}^T \bar{\xi}}{\|\hat{b}\|_2} \geq \frac{1 - \epsilon}{\|\hat{b}\|_2} \geq 1 - \epsilon$ (since $\|\hat{b}\|_2 \leq \|\bar{b}\|_2 = 1$), which also implies that $\|\hat{b}\|_2 \geq 1 - \epsilon$. Since a, \hat{b} , and ξ lie in the same subspace, it follows by the above argument that $\frac{\hat{b}^T \bar{a}}{\|\hat{b}\|_2} \geq (1 - 4\epsilon)$, and hence, $\bar{b}^T \bar{a} = \hat{b}^T \bar{a} \geq (1 - 4\epsilon) \|\hat{b}\|_2 \geq (1 - 4\epsilon)(1 - \epsilon)$, implying the claim. ■

Fact 3.15. *Let $a, b \in \mathbb{R}^r$ be such that $\frac{a^T b}{\|a\|_2 \|b\|_2} \geq 1 - 5\epsilon$ and $\|\hat{b}\|_2 = \lambda \|a\|_2$, where $\lambda \geq 1$ and $\hat{b} := \text{Pj}_a(b)$ is the projection of b on a . Then for any vector $\eta \in \mathbb{R}^r$, it holds that*

$$\|\text{Pj}_\eta(b)\|_2 \geq \lambda \left(\|\text{Pj}_\eta(a)\|_2 - \frac{\sqrt{5\epsilon(2 - 5\epsilon)}}{1 - 5\epsilon} \right) \|a\|_2.$$

Proof. Since the statement is invariant under rotation, we may assume w.l.o.g. that $\eta = \mathbf{1}_j$, the j th-dimensional unit vector in \mathbb{R}^r . Write $b := \hat{b} + \tilde{b}$, where \tilde{b} is the vector orthogonal to a in the subspace spanned by a and b . Then $\|\text{Pj}_\eta(b)\|_2 = b^j$ is the j th component of b , and $\|\text{Pj}_\eta(\hat{b})\|_2 = \hat{b}^j$. Since

$$\|\hat{b}\|_2 \geq \frac{a^T \hat{b}}{\|a\|_2} = \frac{a^T b}{\|a\|_2} \geq (1 - 5\epsilon) \|b\|_2,$$

it follows that

$$\|\tilde{b}\|_2 = \sqrt{\|b\|_2^2 - \|\hat{b}\|_2^2} \leq \sqrt{5\epsilon(2 - 5\epsilon)} \|b\|_2 \leq \frac{\sqrt{5\epsilon(2 - 5\epsilon)}}{1 - 5\epsilon} \|\hat{b}\|_2 = \frac{\lambda \sqrt{5\epsilon(2 - 5\epsilon)}}{1 - 5\epsilon} \|a\|_2.$$

Since $|b^j - \hat{b}^j| = |\tilde{b}^j| \leq \|\tilde{b}\|_2$, and $\hat{b} = \lambda \|a\|_2 \cdot a$, the claim follows. ■

Condition (17) together with Facts 3.13 and 3.14 imply that for any two sets $S, S' \subseteq N^s$, we have $\frac{q_S^T q_{S'}}{\|q_S\|_2 \|q_{S'}\|_2} \geq 1 - 5\epsilon$.

The space partitioning can be done as follows. Let $q_T := \sum_{k \in T} q_k$. We partition the region \mathcal{R}_T into disjoint regions $\mathcal{R}_T(1), \dots, \mathcal{R}_T(h)$, obtained as follows. Let $\mu = \bar{w}_T \cdot \mathbf{1}$, where $\bar{w}_T := \max_{j \in [r]} w_T^j$ (recall the definition of w_T^j from (10)), and define the r -dimensional box $\mathcal{C}_T := \{\nu \in \mathbb{R}_+^r \mid q_T \leq \nu \leq \mu\}$. Note that $\mathcal{R}_T \subseteq \mathcal{C}_T$. We grid the r facets of \mathcal{C}_T that do not contain the point q_T by interlacing equidistant $(r-2)$ -dimensional parallel hyperplanes with inter-separation $\frac{\epsilon \bar{w}_T}{\sqrt{r}}$, for each $j \in [r]$, and with the j th principal axis as their normal. Note that the total number of grid cells obtained is h ; let us call them $\mathcal{C}_1, \dots, \mathcal{C}_h$ (these are $(r-1)$ -dimensional hypercubes). We then define the region $\mathcal{R}(s)$ as the r -dimensional simplex $\mathcal{R}(s) := \text{conv}(\{q_T\} \cup \mathcal{C}_s)$ and $\xi(s) = c(s) - q_T$ as the designated vector, where $c(s)$ is the vertex center of the cell \mathcal{C}_s . Note that the angle condition (17) is satisfied. Indeed, consider any vector q_k such that $q_T + q_k \in \mathcal{R}_T(s)$. Let the ray $\{q_T + \lambda q_k : \lambda \geq 0\}$ hit the boundary cell \mathcal{C}_s in the point x . Consider the triangle formed by the three points $q_T, c(s)$ and x . Then, by construction, the distances between q_T and both $c(s)$ and x are at most \bar{w} , whereas the distance between $c(s)$ and x is at most $\epsilon \bar{w} \sqrt{r}$. It follows that the angle between the two vectors q_k and $\xi(s)$ is no more than $\sin^{-1} \epsilon$, implying (17).

Finally, we define $N^s := \{k \mid q_T + q_k \in \mathcal{R}_T(s)\}$. This gives the required space partitioning of the vectors.

Next, we group the set of items in N into $\ell := 1 + \log_{1+\epsilon} \frac{n}{\epsilon}$ (some possibly empty) *utility-classes* N_1, \dots, N_ℓ , where $N_l = \{k \mid u_k \in [\frac{\epsilon}{n} \cdot B(1+\epsilon)^{l-1}, \frac{\epsilon}{n} \cdot B(1+\epsilon)^l]\}$. Note that for all $k, k' \in N_l$, we have

$$u_k \leq u_{k'}(1 + \epsilon). \quad (18)$$

We can show that for a set of vectors $\{q_k : k \in N^s \cap N_l\}$ that lie in the same region and same utility-group, the *greedy* algorithm that processes the vectors in *non-increasing* order of length gives an $O(\epsilon)$ -optimal solution. For simplicity we assume first that $T = \emptyset$ and $q_T = \mathbf{0}$.

Algorithm 4 GREEDY-COVER($u, \{q_k\}_{k \in N}, C$)

Input: A cost vector $u \in \mathbb{R}_+^N$; accuracy parameter ϵ ; vectors $q_k \in \mathbb{Q}_+^r$ satisfying (18) and (17); a demand $C \in \mathbb{R}_+$;

Output: 9ϵ -optimal solution S to COVER-LIN

- 1: $S' := \min\{u(S) \mid S \subseteq N, |S| \leq \frac{1}{\epsilon}, q_S \text{ is feasible}\}$
 - 2: Order the vectors $q_k, k \in N$, such that $\|q_1\|_2 \geq \|q_2\|_2 \geq \dots$
 - 3: $S \leftarrow \emptyset; k \leftarrow 0$
 - 4: **while** $\|\sum_{k \in S} q_k\|_2 < C$ **do**
 - 5: $k \leftarrow k + 1$
 - 6: $S \leftarrow S \cup \{k\}$
 - 7: **if** $u(S) \leq u(S')$ **then**
 - 8: **return** S
 - 9: **else**
 - 10: **return** S'
-

Lemma 3.16. Consider instance of problem COVER-LIN described by a set of vectors $\{q_k\}_{k \in N}$ satisfying (17) and (18). Then, for any sufficiently small constant $\epsilon > 0$, Algorithm 4 outputs a solution S satisfying $u(S) \leq (1 + 9\epsilon)\text{OPT}$.

Proof. Let S^* be an optimal solution. Since we consider every possible feasible solution of size at most $\frac{1}{\epsilon}$ in step 1, we may assume w.l.o.g. that $|S^*| \geq \frac{1}{\epsilon}$.

We claim that $|S| \leq \frac{|S^*|}{1-5\epsilon} + 1$. To see this claim, let $q_{S^*} := \sum_{k \in S^*} q_k$, and for $k \in N$, denote by $\hat{q}_k := \text{Pj}_{q_{S^*}}(q_k)$ the projection of q_k on q_{S^*} . Let $q_w \in S$ be the last item added to S in step 6 of the algorithm.

Then it is clear that

$$\sum_{k \in S \setminus \{w\}} \|\hat{q}_k\|_2 < \|q_{S^*}\|_2, \quad (19)$$

since otherwise $S' := S \setminus \{w\}$ would satisfy the condition in line 4 (as $\|\sum_{k \in S'} q_k\|_2 \geq \sum_{k \in S'} \|\hat{q}_k\|_2 \geq \|q_{S^*}\|_2 \geq C$, by the feasibility of S^*). By the angle condition (17): $\|\hat{q}_k\|_2 \geq (1 - 5\epsilon)\|q_k\|_2$ for all $k \leq N$. It follows by (19) that

$$\begin{aligned} (1 - 5\epsilon)|S'| \min_{k \in S'} \|q_k\|_2 &\leq (1 - 5\epsilon) \sum_{k \in S'} \|q_k\|_2 \leq \sum_{k \in S'} \|\hat{q}_k\|_2 \\ &< \|q_{S^*}\|_2 = \sum_{k \in S^*} \|\hat{q}_k\|_2 \leq \sum_{k \in S^*} \|q_k\|_2 \leq |S^*| \max_{k \in S^*} \|q_k\|_2. \end{aligned}$$

The claim follows since $\max_{k \in S^*} \|q_k\|_2 \leq \min_{k \in S'} \|q_k\|_2$, by the greedy order in line 2.

By the utility condition (18),

$$u(S) \leq |S| \max_k u_k \leq |S|(1 + \epsilon) \min_k u_k \leq \left(\frac{|S^*|}{1 - 5\epsilon} + 1 \right) (1 + \epsilon) \min_k u_k \leq \left(\frac{1}{1 - 5\epsilon} + \epsilon \right) (1 + \epsilon) u(S^*).$$

The lemma follows. ■

Our QPTAS is given as Algorithm 5. For simplicity, we assume that the algorithm has already a correct guess of the bound B on the value of the optimal solution. After decomposing the instance into classes according to utility and region (steps 2 and 4), the algorithm enumerates over all possible selections of a nonnegative integer $n_{s,l}$ associated to each region s and a utility class l (step 8); this number $n_{s,l}$ represents the largest length vectors that are taken in the potential solution from the set $N^s \cap N_l$. However, for technical reasons, the algorithm does this only for pairs (s, l) for which the set $N^s \cap N_l$ contributes at least $\frac{1}{\epsilon}$ in the optimal solution; the set of pairs that potentially do not satisfy this can be identified by enumeration (steps 6 and 7).

Algorithm 5 COVER-LIN-QPTAS(u, Q, C)

Input: A cost vector $u \in \mathbb{R}_+^N$; accuracy parameter ϵ ; matrix $Q \in \mathcal{CP}_n^*$; a demand $C \in \mathbb{R}_+$

Output: An $O(\sqrt{\epsilon r})$ -optimal solution S to COVER-LIN $\triangleright r = \text{cp-rank}(Q)$

- 1: Obtain sets T and V as explained above and set $N \leftarrow [n] \setminus (T \cup V)$
 - 2: Decompose Q as $Q = UU^T$, where $U \in \mathbb{Q}_+^{n \times r}$
 - 3: Let $\{q_k\}_{k \in N}$ be the columns of U^T corresponding to the indices in N
 - 4: Decompose the set N into space-classes N^1, \dots, N^h and utility classes N_1, \dots, N_ℓ
 - 5: $S \leftarrow T \cup N$ \triangleright Assume instance is feasible
 - 6: **for** each subset of pairs $\mathcal{G} \subseteq [h] \times [\ell]$ **do**
 - 7: **for** each possible selection $(T_{s,l} \subseteq N^s \cap N_l : |T_{s,l}| \leq \frac{1}{\epsilon}, (s, l) \in \mathcal{G})$ **do**
 - 8: **for** each possible selection $(n_{s,l} \in \{1, \dots, |N^s \cap N_l|\} : s \in ([h] \times [\ell]) \setminus \mathcal{G})$ **do**
 - 9: Let $S_{s,l}$ be the set of the $n_{s,l}$ vectors with largest length in $N^s \cap N_l$
 - 10: $S' \leftarrow T \cup \left(\bigcup_{(s,l) \in \mathcal{G}} T_{s,l} \right) \cup \left(\bigcup_{s,l} S_{s,l} \right)$
 - 11: **if** $\|\sum_{k \in S'} q_k\|_2 \geq C$ and $u(S') < u(S)$ **then**
 - 12: $S \leftarrow S'$
 - 13: **return** S
-

Lemma 3.17. For any sufficiently small $\epsilon > 0$ (for instance, $\epsilon < \frac{1}{(r+1)^4}$ for $r \geq 2$), Algorithm 5 runs in time $n^{O((\frac{\sqrt{r}}{\epsilon})^{r+1} \log n)}$ and outputs a solution S satisfying $u(S) \leq (1 + O(\sqrt{\epsilon r})) \text{OPT}$ for any instance of COVER-LIN.

Proof. The running time is obvious since it is dominated by the number of selections in steps 6, 7 and 8, which is at most $\left(\frac{n}{hl}\right)^{O(\frac{h\ell}{\epsilon})} = n^{O((\frac{\sqrt{r}}{\epsilon})^{r+1} \log n)}$.

To see the approximation ratio, fix

$$\lambda := \left(1 - \frac{\sqrt{5\epsilon(2-5\epsilon)r}}{1-5\epsilon}\right)^{-1}. \quad (20)$$

Let S^* be an optimal solution (within N), and for $s \in [h]$ and $l \in [\ell]$, let $S_{s,l}^* := S^* \cap N^s \cap N_l$. Let $\mathcal{G} := \{(s, l) \in [h] \times [\ell] : |S_{s,l}^*| < \frac{1}{\epsilon}\}$ and $T_{s,l} := S^* \cap N^s \cap N_l$, for $(s, l) \in \mathcal{G}$.

Let $\hat{S} := S^* \setminus \left(\bigcup_{(s,l) \in \mathcal{G}} (N^s \cap N_l)\right)$. For $k \in N^s \cap N_l$, let $\hat{q}_k := \text{Pj}_{q_{S_{s,l}^*}}(q_k)$ be the projection of q_k on $q_{\hat{S}_{s,l}} := \sum_{k \in \hat{S}_{s,l}} q_k$. Define the set of pairs

$$\mathcal{H} := \left\{ (s, l) \in [h] \times [\ell] : \left\| \sum_{k \in N^s \cap N_l} \hat{q}_k \right\|_2 \geq \lambda \|q_{\hat{S}_{s,l}}\|_2 \text{ and } |S_{s,l}^*| \geq \frac{1}{\epsilon} \right\}.$$

Then according to Lemma 3.16 (or more precisely, its proof), for every $(s, l) \in \mathcal{H}$ there a choice $n_{s,l} \in \{1, \dots, |N^s \cap N_l|\}$, such that if $S_{s,l}$ is the set of the $n_{s,l}$ vectors with largest length in $N^s \cap N_l$, then $\sum_{k \in S_{s,l}} \|\hat{q}_k\|_2 \geq \lambda \|q_{\hat{S}_{s,l}}\|_2$ and $u(S_{s,l}) \leq \left(\frac{\lambda}{1-5\epsilon} + \epsilon\right) (1+\epsilon) u(\hat{S}_{s,l})$. By this, we can define $p_k = \tau \cdot q_k$,

for $\tau := \frac{\lambda \|q_{\hat{S}_{s,l}}\|_2}{\sum_{k \in S_{s,l}} \|\hat{q}_k\|_2} \leq 1$, such that $\sum_{k \in S_{s,l}} \|\hat{p}_k\|_2 = \lambda \|q_{\hat{S}_{s,l}}\|_2$, where $\hat{p}_k := \text{Pj}_{q_{\hat{S}_{s,l}}}(p_k)$.

Let us now apply Fact 3.15 with $a = a_{s,l} := q_{\hat{S}_{s,l}}$, $b = b_{s,l} := p_{S_{s,l}}$, and $\eta = q_{\hat{S}}$ to get that

$$\|\text{Pj}_{\eta}(b_{s,l})\|_2 \geq \lambda \left(\|\text{Pj}_{\eta}(a_{s,l})\|_2 - \frac{\sqrt{5\epsilon(2-5\epsilon)r}}{1-5\epsilon} \right) \|a_{s,l}\|_2. \quad (21)$$

Summing the above inequalities for all $(s, l) \in \mathcal{H}$, we get

$$\sum_{(s,l) \in \mathcal{H}} \|\text{Pj}_{\eta}(b_{s,l})\|_2 \geq \lambda \left(\sum_{(s,l) \in \mathcal{H}} \|\text{Pj}_{\eta}(a_{s,l})\|_2 - \frac{\sqrt{5\epsilon(2-5\epsilon)r}}{1-5\epsilon} \sum_{(s,l) \in \mathcal{H}} \|a_{s,l}\|_2 \right). \quad (22)$$

Note that $\sum_{(s,l) \in \mathcal{H}} a_{s,l} = \eta$ by construction. By the Cauchy-Schwarz inequality and the nonnegativity of the vectors,

$$\left\| \sum_{(s,l) \in \mathcal{H}} \text{Pj}_{\eta}(a_{s,l}) \right\|_2 = \|\eta\|_2 = \left\| \sum_{(s,l) \in \mathcal{H}} a_{s,l} \right\|_2 \geq \frac{1}{\sqrt{r}} \left\| \sum_{(s,l) \in \mathcal{H}} a_{s,l} \right\|_1 = \frac{1}{\sqrt{r}} \sum_{(s,l) \in \mathcal{H}} \|a_{s,l}\|_1 \geq \frac{1}{\sqrt{r}} \sum_{(s,l) \in \mathcal{H}} \|a_{s,l}\|_2.$$

Using this in (22), we obtain

$$\sum_{(s,l) \in \mathcal{H}} \|\text{Pj}_{\eta}(b_{s,l})\|_2 \geq \lambda \left(1 - \frac{\sqrt{5\epsilon(2-5\epsilon)r}}{1-5\epsilon} \right) \sum_{(s,l) \in \mathcal{H}} \|\text{Pj}_{\eta}(a_{s,l})\|_2 \geq \sum_{(s,l) \in \mathcal{H}} \|\text{Pj}_{\eta}(a_{s,l})\|_2, \quad (23)$$

by our choice of λ . From (22) and $q_{S_{s,l}} \geq b_{s,l}$, it follows by the feasibility of S^* that the solution defined by $S = T \cup \left(\bigcup_{(s,l) \in \mathcal{G}} T_{s,l}\right) \cup \left(\bigcup_{(s,l) \in \mathcal{H}} S_{s,l}\right)$ is feasible.

Clearly, one of the choices in each of the enumeration steps 6, 7, and 8 will capture the above choices \mathcal{G} , $T_{s,l}$ for $(s, l) \in \mathcal{G}$, and $n_{s,l}$, for $(s, l) \in ([h] \times [\ell]) \setminus \mathcal{G}$. It follows the procedure returns a solution S with utility:

$$\begin{aligned}
u(S) &\leq u(T) + \sum_{(s,l) \in \mathcal{G}} u(T_{s,l}) + \sum_{(s,l) \notin \mathcal{G}} u(S_{s,l}) \\
&\leq u(T) + \sum_{(s,l) \in \mathcal{G}} u(T_{s,l}) + \left(\frac{\lambda}{1-5\epsilon} + \epsilon \right) (1+\epsilon) \sum_{(s,l) \notin \mathcal{G}} u(\hat{S}_{s,l}) \\
&\leq \left(\frac{1}{1-5\epsilon - \sqrt{5\epsilon(2-5\epsilon)r}} + \epsilon \right) (1+\epsilon) \left(u(T) + \sum_{(s,l) \in \mathcal{G}} u(T_{s,l}) + \sum_{(s,l) \notin \mathcal{G}} u(\hat{S}_{s,l}) \right) \\
&\leq (1 + O(\sqrt{\epsilon r})) u(S^*).
\end{aligned}$$

The lemma follows. ■

3.4 Dealing with the Approximation in the Decomposition

According to Corollary 2.3, given a matrix Q_i of fixed cp-rank r_i , a decomposition of the form $Q_i := U_i U_i^T - \Delta_i$, where $U_i \in \mathbb{Q}^{r_i \times n}$, $\Delta_i \in \mathbb{R}_+^{n \times n}$, and $\|\Delta_i\|_\infty \leq \delta$, can be done in time $\text{poly}(\mathcal{L}, n^{O(r_i^2)}, \log \frac{1}{\delta})$ for any $\delta > 0$. This leads to a technical issue: if one uses the approximate decomposition $Q_i := U_i U_i^T$ to solve the BQC problem, then the resulting solution can be either infeasible or far from optimal. When $m = O(1)$, this issue can be dealt with at a small loss in the approximation ratio as follows. Let Q_i^{kj} denote the (k, j) th entry of Q_i . Note that, if the k th diagonal element Q_i^{kk} of Q_i is 0, then every entry in the k th row and the k th column of Q_i is 0, since Q_i is positive semi-definite.

We consider the case when f is linear. Let S^* be an optimal solution for PACKING-BQC (resp., COVERING-BQC) and x^* be its characteristic vector. The idea is to show that there is a feasible solution $\tilde{S} \subseteq S^*$ (resp., $\tilde{S} \supseteq S^*$) such that $f(\tilde{S}) \geq (1-\epsilon)f(S^*)$ (resp., $f(\tilde{S}) \leq (1+\epsilon)f(S^*)$), and such that $\tilde{x}^T Q_i \tilde{x} \leq \tilde{C}_i$ (resp., $\tilde{x}^T Q_i \tilde{x} \geq \tilde{C}_i$), where \tilde{x} is the characteristic vector of \tilde{S} and $\tilde{C}_i := C_i - \delta n^2$ (resp., $\tilde{C}_i := C_i + \delta n^2$). Then it would be enough to solve a new instance with the same objective function but with the constraints $x^T U_i U_i^T x \leq C_i$ (resp., $x^T U_i U_i^T x \geq \tilde{C}_i$). Note that \tilde{x} is feasible for the new instance, since $\tilde{x}^T U_i U_i^T \tilde{x} = \tilde{x}^T Q_i \tilde{x} + \tilde{x}^T \Delta_i \tilde{x} \leq \tilde{C}_i + n^2 \delta = C_i$ (resp., $\tilde{x}^T U_i U_i^T \tilde{x} = \tilde{x}^T Q_i \tilde{x} + \tilde{x}^T \Delta_i \tilde{x} \geq \tilde{C}_i$).

Suppose that x is an α -approximate solution for the new instance of PACKING-BQC, then $x^T Q_i x = x^T U_i U_i^T x - x^T \Delta_i x \leq C_i$, since $\Delta_i \geq 0$, and hence x is an $\alpha(1-\epsilon)$ -approximation for the original instance. Similarly, if x is an α -approximate solution for the new instance of COVERING-BQC, then $x^T Q_i x = x^T U_i U_i^T x - x^T \Delta_i x \geq \tilde{C}_i - \delta n^2 = C_i$, and hence x is an $\alpha(1+\epsilon)$ -approximation for the original instance.

The ϵ -approximate solution \tilde{S} can be obtained as follows. Let $\epsilon > 0$ be a fixed constant, and define $\lambda = \frac{m}{\epsilon}$. Let \mathcal{U} be the λ items of highest utility in the optimal solution S^* . This gives a partition of $[n]$ into three sets $\mathcal{U} \cup \mathcal{V} \cup N$, as described in Section 3.1. We can make the following assumptions w.l.o.g.:

- (A1) $|S^*| \geq \frac{m}{\epsilon}$;
- (A2) for each $i \in [m]$ the set $\kappa_i := \{k \in S^* \cap N : Q_i^{kk} > 0\} \neq \emptyset$ (resp., $\kappa_i := \{k \in N \setminus S^* : Q_i^{kk} > 0\} \neq \emptyset$).

(A1) can be assumed since otherwise the algorithm that enumerates over all possible sets of size at most $\frac{m}{\epsilon}$ is optimal. (A2) can be assumed since if it is violated by $i \in [m]$, then after fixing the set \mathcal{U} (resp, after fixing the set \mathcal{U} and fixing the variables in the set $\{k \in N : Q_i^{kk} = 0\}$ to 1), the i th constraint becomes redundant. Thus the algorithm that enumerates all subsets \mathcal{U} of size at most $\frac{m}{\epsilon}$ and all subsets of constraints, removing a subset at a time (resp., enumerates all subsets \mathcal{U} of size at most $\frac{m}{\epsilon}$ and all subsets of constraints, removing a subset at a time and fixing the variables in the set $\{k \in N : Q_i^{kk} = 0\}$ to 1), can assume an instance satisfying (A2).

Then, assuming (A1) and (A2), for each $i \in [m]$, we pick an index $k(i) \in \kappa_i$ and set the corresponding variable to 0 (resp., to 1). This yields an ϵ -optimal solution \tilde{S} , whose characteristic vector x also satisfies $x^T Q_i x \leq (x^*)^T Q_i x^* - Q_i^{k(i), k(i)} \leq \tilde{C}_i$ (resp., $x^T Q_i x \geq (x^*)^T Q_i x^* + Q_i^{k(i), k(i)} \geq \tilde{C}_i$), if we choose $\delta := \frac{\min_{i,k} Q_i^{kk}}{n^2}$.

4 Applications of The Approximation Scheme

4.1 A PTAS for Multi-Objective Optimization

In this section we study optimization problem with multiple objectives. In such a problem it is concerned with optimizing more than one objective function simultaneously. In fact, we consider the following binary multi-objective quadratic problem (BMOQ):

$$\begin{aligned} \text{(BMOQ)} \quad & \{\max f(x), \min g(x)\} \\ \text{subject to} \quad & x \in X \subseteq \{0, 1\}^n \end{aligned}$$

where $f(x) = \{f_i(x)\}_{i \in \mathcal{I}}$ and $g(x) = \{g_j(x)\}_{j \in \mathcal{J}}$, and

$$X = \{x \in \{0, 1\}^n \mid x^T Q_i x + q_i^T x \leq C_i^2; Ax \leq b; i \in [m]\},$$

where $Q_i \in \mathcal{CP}_n^*$, $q_i \in \mathbb{R}_+^n$ for all $i \in [m]$, $A \in \mathbb{R}_+^{d \times n}$, $b \in \mathbb{R}_+^d$, and m, d are constant.

Typically, a feasible solution that simultaneously optimizes each objective may not exist due to the trade-off between the different objectives. This issue can be captured by the notion of *Pareto-optimal frontier* (or *Pareto curve*), which is a set of all feasible solutions whose vector of the objectives is not *dominated* by any other one. Unfortunately, the size of such a Pareto-optimal frontier set is often exponential in size of the input. Hence, a natural goal in this setting is, given an instance of a (BMOQ) problem, to efficiently achieve an ϵ -approximation of the Pareto optimality that consists of a number of solutions that is polynomial in size of the instance, for every constant $\epsilon > 0$. Formally, let I be an instance of the multi-objective optimization problem (BMOQ), consider the following definitions taken from [MS13b]:

Definition 4.1. A *Pareto-optimal frontier* P of I is a subset of X , such that for any solution $S \in P$ there is no solution $S' \in X$ such that $f_i(S') \geq f_i(S)$ for all $i \in \mathcal{I}$ and $g_j(S') \leq g_j(S)$ for all $j \in \mathcal{J}$, with strict inequality for at least one of them.

Definition 4.2. For $\epsilon > 0$, an ϵ -approximate Pareto-optimal frontier of I , denoted by P_ϵ , is a set of solutions, such that for all $S \in X$, there exists a solution $S' \in P_\epsilon$ such that $f_j(S') \geq (1 - \epsilon)f_j(S)$ and $g_j(S') \leq g_j(S)/(1 - \epsilon)$ for all $i \in \mathcal{I}, j \in \mathcal{J}$.

Computing an (approximate) Pareto-optimal frontier has been known to be a very efficient approach in designing approximation schemes for combinatorial optimization problems with multiple objectives [EKP02, PY00, MS13b, MS13a]. Erlebach et al. [EKP02] give a PTAS for multi-objective multi-dimensional knapsack problem. In this paper, we will extend their method to the case with quadratic constraints. Let I be an instance of the BMOQ problem, and denote $f_j(x) = a_j^T x$, $g_j(x) = b_j^T x$ for all $j \in [p]$. Let $\epsilon > 0$ be a fixed constant. The Algorithm PARETO-OPT below will produce a set P_ϵ , which is an ϵ -approximate Pareto-optimal frontier to the instance I , in polynomial time. Moreover, the size of P_ϵ is guaranteed to be also polynomial in size of the input for every fixed $\epsilon > 0$.

We first compute ϵ^2 -approximate solutions S_j, T_j to the maximization problem with single objective functions $f_j = a_j^T x$, $g_j = b_j^T x$ subject to $x \in X$, respectively, by using an extension of Algorithm 1. As a result, for any feasible solution S , we have $f_j(S) \leq (1 - \epsilon^2)^{-1} f_j(S_j)$ and $g_j(S) \leq (1 - \epsilon^2)^{-1} g_j(T_j)$ for all $j \in [p]$. For each $j \in [p]$, we consider $v_j + 2$ (lower) bounds $\{\alpha_{j\lambda_j}, 0 \leq \lambda_j \leq v_j + 1\}$ for the objective f_j and $w_j + 3$ (upper) bounds $\{\beta_{j\eta_j}, 0 \leq \eta_j \leq w_j + 2\}$ for the objective g_j . We define the set $\Lambda \times \Gamma \subseteq \mathbb{R}_+^{2p-1}$, which

contains all the tuples $(\lambda_1, \dots, \lambda_{p-1}, \eta_1, \dots, \eta_p)$. Note that the size of $\Lambda \times \Gamma$ is bounded by a polynomial in size of the input since p is constant and v_j, w_j are also bounded by a polynomial in size of the input. The main idea of the algorithm is that, for each tuple of the bounds, we try to maximize the last objective f_p subject to $f_j \geq \alpha_{j\lambda_j}$, for all $j \in [p-1]$, and $g_j \leq \beta_{j\eta_j}$, for all $j \in [p]$. To do that, we consider all possibility of choosing a subset of $[n]$ of at most (a constant number) λ of items. Denote Σ as the set of all such subsets and let $\mathcal{X} = \Sigma^p$. For each tuple $(X_1, \dots, X_p) \in \mathcal{X}$, where X_j is a set of items of highest utility with respect to f_i , we construct a tuple (Y_1, \dots, Y_p) such that Y_j is the set of all items that will not be put into the knapsack given that items in X_j are the ones with highest utility with respect to f_j in the solution. Now for each tuple $(X_1, \dots, X_p, Y_1, \dots, Y_p)$ such that $\bigcup_{j \in [p]} (X_j \cap Y_j) = \emptyset$, we solve the convex program (CP2[\mathcal{U}]) and obtain (if exists) an ϵ -optimal solution x^* :

$$\begin{aligned}
(\text{CP2}[\mathcal{U}]) \quad & \max a_p^T x \\
\text{subject to} \quad & x^T Q_i x + q_i^T x \leq C_i^2, i \in [m], \\
& Ax \leq b, \\
& a_j^T x \geq \alpha_{j\lambda_j}, j \in [p-1], \\
& b_j^T x \leq \beta_{j\eta_j}, j \in [p], \\
& x_k = 1, \text{ for } k \in \mathcal{U}, x_k = 0, \text{ for } k \in \mathcal{V}, \\
& x_k \in [0, 1], \text{ for } k \in N.
\end{aligned}$$

Define $t_i := U_i^T[*; N]x_N^*$, $t'_i := \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; N]x_N^*$, $q_i^T[*; N]x_N^* = v_i$ for all $i \in [m]$, $A[*; N]x_N^* = \sigma$, $a_j^T[*; N]x_N^* = \theta_j$ for $j \in [p-1]$, and $b_j^T[*; N]x_N^* = \theta'_j$ for $j \in [p]$. We define a polytope $\mathcal{P}(\mathcal{U}) \subseteq [0, 1]^N$ as follows:

$$\mathcal{P}(\mathcal{U}) = \left\{ y \in [0, 1]^N \left| \begin{array}{l} U_i^T[*; N]y \leq t_i, \mathbf{1}_{\mathcal{U}}^T Q_i[\mathcal{U}; N]y \leq t'_i, q_i^T[*; N]y \leq v_i, \text{ for } i \in [m] \\ A[*; N]y \leq \sigma \\ a_j^T[*; N]y \geq \theta_j \text{ for } j \in [p-1], b_j^T[*; N]y \leq \theta'_j \text{ for } j \in [p] \end{array} \right. \right\}$$

We can find (if exists) a (BFS) y in this polytope such that $a_p^T y \geq a_p^T x_N^*$. By rounding down this fractional solution y and setting $x_k \in \{0, 1\}$ according to the assumption $k \in \mathcal{U} \cup \mathcal{V}$, we obtain an integral solution $\bar{x} \in \{0, 1\}^N$.

There are totally $n^{p\lambda}$ tuples of the form $(X_1, \dots, X_p, Y_1, \dots, Y_p)$. For each such tuple such that the condition in Step 18 is satisfied and (CP1[\mathcal{U}]) is feasible, the algorithm outputs an integral solution \bar{x} . All possible integral solutions obtained in this way are collected in the set P_ϵ . This completes the description of the algorithm.

Theorem 4.3. *Let I be an instance of the BMOQ problem. For every fixed $\epsilon > 0$, the Algorithm 6 runs in polynomial time in size of the input and produces an ϵ -approximate Pareto-optimal frontier P_ϵ for I .*

Proof. It is not difficult to see that the running time of Algorithm 6 is polynomial in size of the input as the parameters $\epsilon, \epsilon', \mu, \lambda$ are constants. Hence, we need only to show the approximation factor of the algorithm. More precisely, let S be an arbitrary feasible solution, we need to prove that there exists a solution $S' \in P_\epsilon$ such that $f_j(S') \geq (1 - \epsilon)f_j(S)$ and $g_j(S') \leq g_j(S)/(1 - \epsilon)$, for all $j \in [p]$. Let us define

$$\lambda_j = \max\{\lambda | \alpha_{j\lambda} \leq f_j(S)\} \quad \text{and} \quad \eta_j = \min\{\eta | \beta_{j\eta} \geq g_j(S)\}.$$

It follows that

$$\alpha_{j\lambda_j} \leq f_j(S) \leq \alpha_{j\lambda_j}/(1 - \epsilon^2) \quad \text{and} \quad \beta_{j\eta_j} \geq g_j(S) \geq (1 - \epsilon^2)\beta_{j\eta_j}$$

Let $\delta = \min\{\lambda, |S|\}$. For each $j \in [p]$, let X_j be the set that contains δ items in S with largest utility with respect to the objective f_j . One can see that S is a feasible solution to (CP2[\mathcal{U}]) with the tuple

Algorithm 6 PARETO-OPT($\{a_j^T, b_j^T, Q_i, C_i\}_{j \in [p], i \in [m]}, \epsilon$)

Input: Utilities, demand vectors and capacities ($\{a_j^T, b_j^T, Q_i, C_i\}_{j \in [p], i \in [m]}$); accuracy parameter ϵ

Output: $(1 - \epsilon)$ -approximate Pareto optimality P_ϵ

```

1:  $P_\epsilon \leftarrow \emptyset$ 
2: Decompose  $Q_i$  into  $U_i^T U_i$ , where  $U_i$  has  $r_i$  rows, for all  $i \in [m]$   $\triangleright r_i = \text{cp-rank}(Q_i)$ 
3: for  $j \in [p]$  do
4:    $S_j \leftarrow \text{PTAS}(a_j^T, Q_i, C_i, i \in [m], \epsilon^2)$   $\triangleright$  Find an  $\epsilon^2$ -approximate solution to  $f_j$ 
5:    $T_j \leftarrow \text{PTAS}(b_j^T, Q_i, C_i, i \in [m], \epsilon^2)$   $\triangleright$  Find an  $\epsilon^2$ -approximate solution to  $g_j$ 
6:    $v_j \leftarrow \lceil \log_{(1-\epsilon')^{-1}} f_j(S_j) \rceil$ ;  $w_j \leftarrow \lceil \log_{(1-\epsilon')^{-1}} g_j(T_j) \rceil$ 
7:    $\alpha_{j0} \leftarrow 0$ ;  $\alpha_{j\lambda_j} \leftarrow (1 - \epsilon^2)^{1-\lambda_j}$  for all  $\lambda_j \in [v_j + 1]$ 
8:    $\beta_{j0} \leftarrow 0$ ;  $\beta_{j\eta_j} \leftarrow (1 - \epsilon^2)^{1-\eta_j}$  for all  $\eta_j \in [w_j + 2]$ 
9:    $\Lambda \leftarrow [0, v_1] \times \dots \times [0, v_{p-1}]$ ;  $\Gamma \leftarrow [0, w_1] \times \dots \times [0, w_p]$ 
10:   $\phi \leftarrow \sum_{i=1}^m (r_i + 1)$ ;  $\lambda \leftarrow \left\lceil \frac{(\phi + 2p + d + m - 1)(1 + \epsilon)}{\epsilon(1 - \epsilon)} \right\rceil$ 
11:   $\mathcal{Y} \leftarrow \emptyset$ ;  $\mathcal{X} \leftarrow \{(X_1, \dots, X_p) \mid X_j \subseteq [n], |X_j| \leq \lambda, j \in [p]\}$ 
12:  for each  $(X_1, \dots, X_p) \in \mathcal{X}$  do
13:    for  $j \in [p]$  do
14:       $Y_j \leftarrow \{k \in [n] \setminus X_j \mid a_{jk} > \min\{a_{jk'} \mid k' \in X_j\}\}$ 
15:     $\mathcal{Y} \leftarrow \mathcal{Y} \cup (Y_1, \dots, Y_p)$ 
16:  for each tuple  $(\lambda_1, \dots, \lambda_{p-1}, \eta_1, \dots, \eta_p) \in \Lambda \times \Gamma$  do
17:    for each tuple  $(X_1, \dots, X_p, Y_1, \dots, Y_p) \in \mathcal{X} \times \mathcal{Y}$  do
18:      if  $\mathcal{U} \cap \mathcal{V} = \emptyset$  then
19:        Find (if exists) an  $\epsilon$ -optimal solution  $x^*$  to the convex program (CP2[ $\mathcal{U}$ ])
20:        Define a polytope  $\mathcal{P}(\mathcal{U}) \subseteq [0, 1]^N$ 
21:        Find an BFS  $y$  of  $\mathcal{P}(\mathcal{U})$  such that  $a_p^T y \geq a_p^T x_N^*$ 
22:         $\bar{x} \leftarrow \{(\bar{x}_1, \dots, \bar{x}_n) \mid \bar{x}_k = \lfloor y_k \rfloor \text{ for } k \in N, x_k = 1, \text{ for } k \in \mathcal{U}, x_k = 0, \text{ for } k \in \mathcal{V}\}$   $\triangleright$ 
        rounding down solution  $y$ 
23:         $P_\epsilon \leftarrow P_\epsilon \cup \bar{x}$ 
24: return  $P_\epsilon$ 

```

$(\lambda_1, \dots, \lambda_{p-1}, \eta_1, \dots, \eta_p)$ and the tuple (X_1, \dots, X_p) . Let y be a BFS to the polytope $\mathcal{P}(\mathcal{U})$ and let \bar{x} be the corresponding rounded solution. By the same arguments provided as in the section 26, one can easily proof that \bar{x} is feasible. Denote S' by set of items corresponding to the integral solution \bar{x} . We have:

$$g_j(S') = b_j^T \bar{x} \leq \beta_{j\eta_j} \leq g_j(S)/(1 - \epsilon^2) \leq g_j(S)/(1 - \epsilon) \quad (24)$$

Furthermore, for every $j \in [p - 1]$:

$$a_j^T x^* \geq \alpha_{j\lambda_j} \geq (1 - \epsilon^2) f_j(S),$$

and for $j = p$, we even have $a_p^T x^* \geq (1 - \epsilon) \text{opt} \geq (1 - \epsilon) f_p(S)$, because x^* is ϵ -optimal to (CP2[\mathcal{U}]). We consider two cases below.

Case I: If $\delta = |S|$, (i.e. $|S| \leq \lambda$), the set S' will contain S as a subset. Hence, we have $f_j(S') \geq f_j(S)$ for all $j \in [p]$.

Case II: If $\delta = \lambda$, we prove that $f_j(S') \geq (1 - \epsilon) f_j(S)$ for all $j \in [p]$. Let a_{jk} be the smallest utility among the λ items with highest utility in S with respect to objective f_j , we have

$$a_j^T y + a_j^T \mathbf{1}_{\mathcal{U}} \geq a_j^T x_N^* + a_j^T \mathbf{1}_{\mathcal{U}} = a_j^T x^* \geq (1 - \epsilon) \text{OPT} \geq (1 - \epsilon) \lambda a_{jk}$$

On the other hand, since \bar{x} is obtained by rounding down y , and the fact that y has at most $\phi + 2p + d + m - 1$ fractional components, we have

$$f_j(S') = a_j^T \bar{x} = a_j^T \bar{x}_N + a_j^T \mathbf{1}_{\mathcal{U}} \geq a_j^T y - a_{jk}(\phi + 2p + d + m - 1) + a_j^T \mathbf{1}_{\mathcal{U}}$$

Therefore,

$$\begin{aligned} f_j(S') &\geq a_j^T x^* - \frac{\phi + 2p + d + m - 1}{\lambda} \lambda a_{jk} \geq a_j^T x^* - \frac{\phi + 2p + d + m - 1}{\lambda(1 - \epsilon)} a_j^T x^* \\ &= (1 - \frac{\phi + 2p + d + m - 1}{\lambda(1 - \epsilon)}) a_j^T x^* \\ &\geq (1 - \frac{\epsilon}{1 + \epsilon}) a_j^T x^* = \frac{a_j^T x^*}{1 + \epsilon} \end{aligned}$$

Note that $a_j^T x^* \geq (1 - \epsilon^2) f_j(S)$, we have

$$f_j(S') \geq \frac{a_j^T x^*}{1 + \epsilon} \geq \frac{1 - \epsilon^2}{1 + \epsilon} f_j(S) = (1 - \epsilon) f_j(S). \quad (25)$$

From (24) and (25), the proof is completed. \blacksquare

4.2 Binary Quadratically Constrained Quadratic Programming Problems

We consider the binary quadratically constrained quadratic programming problem which takes the form:

$$\begin{aligned} (\text{BQCQP}) \quad & \max w(x) = x^T Q x + q^T x \\ \text{subject to} \quad & x \in X \subseteq \{0, 1\}^n, \end{aligned}$$

where $Q \in \mathcal{CP}_n^*$, $q \in \mathbb{R}_+^n$, and the feasible region X is defined as in the previous section.

Theorem 4.4. *The problem (BQCQP) admits a PTAS.*

Proof. Let I be an instance of (BQCQP) and let $\epsilon > 0$ be a fixed constant. Since $p = \text{cp-rank}(Q)$ is fixed, one can find in polynomial time a factorization $Q = \sum_{i=1}^p a_i a_i^T$, $a_i \in \mathbb{R}_+^n$. Hence, the objective function can be written in the form $\sum_{j=1}^p (a_j^T x)^2 + a_{p+1}^T x$, where $a_{p+1} = q$. Applying the Theorem 4.3 for the multi-objective instance with $|\mathcal{I}| = p + 1, |\mathcal{J}| = 0$, we obtain P_ϵ as an approximation of the Pareto-optimal frontier. Let $x^* \in X$ be an optimal solution to the instance I , there exists a solution $y \in P_\epsilon$ such that $a_j^T y \geq (1 - \epsilon) a_j^T x^*$, for all $j \in [p + 1]$. Hence,

$$w(y) = \sum_{j=1}^p (a_j^T y)^2 + a_{p+1}^T y \geq (1 - \epsilon)^2 \sum_{j=1}^p (a_j^T x^*)^2 + (1 - \epsilon) a_{p+1}^T x^* \geq (1 - \epsilon)^2 \text{OPT}.$$

Let $y^* = \arg\max\{w(x) \mid x \in P_\epsilon\}$ (note that the size of P_ϵ is polynomial in size of the input), we have $w(y^*) \geq w(y) \geq (1 - \epsilon)^2 \text{OPT}$. \blacksquare

4.3 Binary Multiplicative Programming Problems

We consider the following problem where the objective is the product of a number of linear functions.

$$\begin{aligned} (\text{BMPP}) \quad & \max w(x) = (a_1^T x) \cdot (a_2^T x) \cdot \dots \cdot (a_p^T x) \\ \text{subject to} \quad & x \in X \subseteq \{0, 1\}^n, \end{aligned}$$

where $a_j \in \mathbb{R}_+^n$ for all $j \in [p]$.

Theorem 4.5. *If p is constant, the problem (BMPP) admits a PTAS.*

Proof. Again, applying the Theorem 4.3 for the multi-objective instance with $|\mathcal{I}| = p, |\mathcal{J}| = 0$, we obtain an approximation of the Pareto set P_ϵ . Let $x^* \in X$ be an optimal solution to the instance of (BMPP), there exists a solution $y \in P_\epsilon$ such that $a_j^T y \geq (1 - \epsilon)a_j^T x^*$, for all $j \in [p]$. Hence,

$$w(y) = \prod_{j=1}^p a_j^T y \geq (1 - \epsilon)^p \prod_{j=1}^p (a_j^T x^*) = (1 - \epsilon)^p \text{OPT}.$$

Again, let $y^* = \text{argmax}\{w(x) \mid x \in P_\epsilon\}$, it follows $w(y^*) \geq (1 - \epsilon)^p \text{OPT}$. Note that p is constant, the result follows. \blacksquare

4.4 Sum of Ratios

We consider the binary quadratically constrained programming with a rational objective function.

$$\begin{aligned} \text{(SUM-RATIO)} \quad & \max w(x) = \sum_{j=1}^p \frac{a_j^T x}{b_j^T x} \\ \text{subject to} \quad & x \in X \subseteq \{0, 1\}^n. \end{aligned}$$

This problem belongs to the class of fractional programming which has important applications in several areas such as transportation, finance, engineering, statistics (see the survey paper by [SS03] and the references therein, for more applications).

Theorem 4.6. *Suppose that p is fixed. The problem (SUM-RATIO) admits a PTAS.*

Proof. applying the Theorem 4.3 for the instance with $|\mathcal{I}| = |\mathcal{J}| = p$, we obtain an approximation of the Pareto set P_ϵ of the instance of multi-objective problem. Let $x^* \in X$ be an optimal solution to an instance of (SUM-RATIO) problem, there exists a solution $y \in P_\epsilon$ such that $a_j^T y \geq (1 - \epsilon)a_j^T x^*$ and $b_j^T y \leq b_j^T x^*/(1 - \epsilon)$, for all $j \in [p]$. Hence,

$$w(y) = \sum_{j=1}^p \frac{a_j^T y}{b_j^T y} \geq (1 - \epsilon)^2 \sum_{j=1}^p \frac{a_j^T x^*}{b_j^T x^*} = (1 - \epsilon)^2 \text{OPT}.$$

Again, let $y^* = \text{argmax}\{w(x) \mid x \in P_\epsilon\}$, the proof is completed. \blacksquare

5 Conclusion

In this paper, we have investigated the approximability of binary quadratic programming problems when the cp-rank of the constraints' matrices are bounded by a constant. It turns out that bounding the cp-rank of these matrices makes several interesting variants of the quadratic programming problem tractable. In particular, our results hint that limiting the cp-rank makes the quadratic problem exhibit similar approximability behavior as the linear case, assuming a constant number of packing or covering quadratic constraints. For the case with any number of quadratic constraints and linear objective function, one can easily obtain a $O(m\sqrt{r})$ -approximation algorithm for the packing problem, where $r = \max_{i \in [m]} r_i$. The first interesting question is whether there exists a (Q)PTAS for the covering problem with a constant number of quadratic constraints. Extending our results to the case of low nonnegative ranks, and finding non-trivial inapproximability bounds, parametrized by the cp-rank of the constraints matrices are also another interesting open problems.

References

- [AFLS01] K. Allemand, K. Fukuda, T. Liebling, and E. Steiner. A polynomial case of unconstrained zero-one quadratic optimization. *Math. Program.*, 91(1):49–52, 2001.
- [AGKM12] S. Arora, R. Ge, R. Kannan, and A. Moitra. Computing a nonnegative matrix factorization - provably. In *STOC*, pages 145–162, 2012.
- [AKA94] B. Alidaee, G. Kochenberger, and A. Ahmadian. 0-1 quadratic programming approach for the optimal solution of two scheduling problems. *International Journal of Systems Science*, 25:401–408, 1994.
- [BPR96] S. Basu, R. Pollack, and M. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *J. ACM*, 43(6):1002–1045, 1996. Preliminary version in *FOCS* 1994.
- [BSM03] A. Berman and N. Shaked-Monderer. *Completely Positive Matrices*. Singapore: World Scientific, 2003.
- [Bur09] S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Math. Program.*, 120(2):479–495, 2009.
- [CEK14] C. Chau, K. Elbassioni, and M. Khonji. Truthful mechanisms for combinatorial ac electric power allocation. In *AAMAS*, pages 1005–1012, 2014.
- [Cha93] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete and Computational Geometry*, 10(1):377–409, 1993.
- [CHW76] A. Chandra, D. Hirschberg, and C. Wong. Approximate algorithms for some generalized knapsack problems. *Theor. Comput. Sci.*, 3(3):293–304, 1976.
- [DG14] P. Dickinson and L. Gijben. On the computational complexity of membership problems for the completely positive cone and its dual. *Computational Optimization and Applications*, 57(2):403–415, 2014.
- [Dug09] S. Dughmi. Submodular functions: Extensions, distributions, and algorithms. a survey. *CoRR*, abs/0912.0322, 2009.
- [EKP02] T. Erlebach, H. Kellerer, and U. Pferschy. Approximating multiobjective knapsack problems. *Management Science*, 48(12):1603–1612, 2002.
- [FC84] A. Frieze and M. Clarke. Approximation algorithms for the m -dimensional 0-1 knapsack problem: worst-case and probabilistic analyses. *European Journal of Operational Research*, 15:100–109, 1984.
- [GHS80] G. Gallo, P. Hammer, and B. Simeone. Quadratic knapsack problems. *Math. Program.*, 12:132–149, 1980.
- [GKR11] V. Goyal, L. Kaya, and R. Ravi. An fptas for minimizing the product of two non-negative linear cost functions. *Math. Program.*, 126(2):401–405, 2011.
- [GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, New York, 1988.
- [GR13] V. Goyal and R. Ravi. An fptas for minimizing a class of low-rank quasi-concave functions over a convex set. *Oper. Res. Lett.*, 41(2):191–196, 2013.

- [GV88] D. Grigoriev and N. Vorobjov. Solving systems of polynomial inequalities in subexponential time. *J. Symb. Comput.*, 5(1/2):37–64, 1988.
- [Hal86] M. Hall. *Combinatorial Theory (2nd Ed.)*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
- [HXZ⁺11] Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki. Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering. *IEEE Transactions on Neural Networks*, 22(12):2117–2131, 2011.
- [IK75] O. Ibarra and C. Kim. Fast approximation algorithms for the knapsack and sum of subset problems. *J. ACM*, 22(4):463–468, 1975.
- [IP01] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.
- [JW02] D. Rader Jr. and G. Woeginger. The quadratic 0-1 knapsack problem with series-parallel support. *Oper. Res. Lett.*, 30(3):159–166, 2002.
- [KN07] J. Kelner and E. Nikolova. On the hardness and smoothed complexity of quasi-concave minimization. In *FOCS*, pages 472–482, 2007.
- [KS10] H. Kellerer and V. Strusevich. Fully polynomial approximation schemes for a symmetric quadratic knapsack problem and its scheduling applications. *Algorithmica*, 57(4):769–795, 2010.
- [KS12] H. Kellerer and Vitaly A. Strusevich. The symmetric quadratic knapsack problem: approximation and scheduling applications. *4OR*, 10(2):111–161, 2012.
- [KST09] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *SODA*, pages 545–554, 2009.
- [KW07] W. Kern and G. Woeginger. Quadratic programming and combinatorial minimum weight product problems. *Math. Program.*, 110(3):641–649, 2007.
- [Lau70] D. Laughhunn. Quadratic binary programming with application to capital-budgeting problems. *Operations Research*, 18:454–461, 1970.
- [LMNS09] J. Lee, V. Mirrokni, V. Nagarajan, and M. Sviridenko. Non-monotone submodular maximization under matroid and knapsack constraints. In *STOC*, pages 323–332, 2009.
- [McM70] P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17(02):179–184, 1970.
- [Moi13] A. Moitra. An almost optimal algorithm for computing nonnegative rank. In *SODA*, pages 1454–1464, 2013.
- [MS13a] S. Mittal and A. Schulz. An fptas for optimizing a class of low-rank functions over a polytope. *Math. Program.*, 141(1-2):103–120, 2013.
- [MS13b] S. Mittal and A. Schulz. A general framework for designing approximation schemes for combinatorial optimization problems with many objectives combined into one. *Operations Research*, 61(2):386–397, 2013.
- [MY80] R. Mcbride and J. Yormark. An implicit enumeration algorithm for quadratic integer programming. *Manage. Sci.*, 26(3):282296, 1980.
- [NT08] A. Nemirovski and M. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 5 2008.

- [Pis07] D. Pisinger. The quadratic knapsack problem - a survey. *Discrete Applied Mathematics*, 155(5):623–648, 2007.
- [PS13] U. Pferschy and J. Schauer. Approximating the quadratic knapsack problem on special graph classes. In *WAOA*, pages 61–72, 2013.
- [PY00] C. Papadimitriou and M. Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *FOCS*, pages 86–92, 2000.
- [Ren92] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals. *J. Symb. Comput.*, 13(3):255–352, 1992.
- [Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, New York, 1986.
- [SS03] S. Schaible and J. Shi. Fractional programming: the sum-of-ratios case. *Optimization Methods and Software*, 18:219–229, 2003.
- [Svi04] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, 2004.
- [Vav09] S. Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 20(3):1364–1377, 2009.
- [VCZ11] J. Vondrák, C. Chekuri, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *STOC*, pages 783–792, 2011.
- [VHVD08] A. Vandendorpe, D. Ho, S. Vanduffel, and P. Dooren. On the parameterization of the creditrisk⁺ model for estimating credit portfolio risk. *Insurance: Mathematics and Economics*, 42(2):736–745, 2008.
- [Von08] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *STOC*, pages 67–74, 2008.
- [Xu12] Z. Xu. A strongly polynomial fptas for the symmetric quadratic knapsack problem. *European Journal of Operational Research*, 218(2):377–381, 2012.
- [YC13] L. Yu and C. Chau. Complex-demand knapsack problems and incentives in ac power systems. In *AAMAS*, pages 973–980, 2013.
- [ZS05] R. Zass and A. Shashua. A unifying approach to hard and probabilistic clustering. In *ICCV*, pages 294–301, 2005.